アルゴリズム学習のための アプリケーションソフト

太　田　幸　一

要旨

本学の講義「アルゴリズム論」において，プログラミングに慣れ親しみ，それと同時に学習の効果を高めるため制作した，アプリケーションソフトを紹介する。本論文では，アプリケーションソフトの説明と使用方法を解説する。アプリケーションソフトは，私が独自で設計・作成した，流れ図によるプログラム言語のシステムで，その操作はマウスによって一種のプログラムが作成でき，その実行が行えるソフトである。このアプリケーションプログラムの目的は，プログラムの基本となる連接・繰り返し・判断・関数（サブルーチン）の学習である。

キーワード：アルゴリズム，流れ図，プログラム言語，情報処理教育

１．は　じ　め　に

最初の，流れ図によるアルゴリズム学習のためのアプリケーションソフトを使用したアルゴリズムの授業では，1995年に出版した「構造化プログラミングのアルゴリズム」を教科書として使用した。その書籍には，3.5インチのフロッピーディスクが付いていて，そのフロッピーディスクの中に，このアプリケーションソフトが入っている。

アルゴリズムの授業の受講者は，その教科書のアプリケーションソフトを操作しながら，教科書の例題や問題を解いていくことで，アルゴリズムを習得していった。

書籍の出版時以降，アルゴリズムの授業には，アルゴリズムすなわちプログラミングの学習の補助材料として，当アプリケーションソフトを必ず使用してきた。

長年に渡っての使用であるため，いくども修正や改良などを行い続けた。

したがって，書籍に添付していた初期バージョンのアプリケーションソフトは，数年で古くなったことで，そのソフトに取って代わって，私が，改良したその時々のバージョンアップ版を授業で提供し，参考書も従来の書籍から，ブラウザで表示することのできるネット公開による e-learning 教材に代えて活用している。

本論文では述べないが，最近ではこのアプリケーションソフトに加えて，そのソフトのシステムを利用したパズルゲームを考案・作成し，最初の数回で行う導入部の授業で提供し，学習者がいっそう楽しく学べる環境となるように工夫した。

流れ図によるアルゴリズム学習のためのアプリケーションソフトと，そのアプリケーションソフトを使用した授業の内容を例示とともに紹介する。

最後に，それらの授業を通してのアンケート調査により，プログラミング言語を使わない，本アプリケーションによるアルゴリズム学習に対する効果を調べた。

２．アプリケーションソフトの初期画面

現在，「アルゴリズム論」や「アルゴリズム実習」など，私が担当するアルゴリズム関連の授業の中では，マウスで流れ図記号を，移動・削除・複写することによって簡単に流れ図が作成でき，しかもその流れ図に従って計算などが実行できる，私独自で設計・作成したアプリケーションソフトを使って，講義および実習を行っている。

この「logic master」という名のアプリケーションソフトは，授業に役立てる目的で開発したものである。

「logic master」の操作は，パソコン画面上に縦に並んだ JIS で定義されている流れ図記号のアイコンから，マウスのクリックで一つ選び出し，マウスによる流れ図記号のアイコン移動と，マウスのクリックで流れ図の中にその流れ図記号を挿入していくことによって，簡単に流れ図を構築していくのである。

したがって「logic master」は，流れ図の作成，すなわちコンピュータ・プログラムに対する論理思考の組み立てを，視覚的に，しかも簡便・手軽に操作できる仕組みになっている。

図１は，Windows 7 における「logic master」の初期画面である。

「logic master」内の「フローチャート表示ウィンドウ」中の左端に，流れ図（フローチャート）の部品である流れ図記号を立体的に表したアイコンが，縦一列に並んでいる。

縦に並んでいるそれらの流れ図記号は，表１の JIS（日本工業規格）で定めた記号の形になっている。ソフトで使われる記号の役割は，次ページの表１の「記号の説明」に書か

図１　「logic master」の初期画面

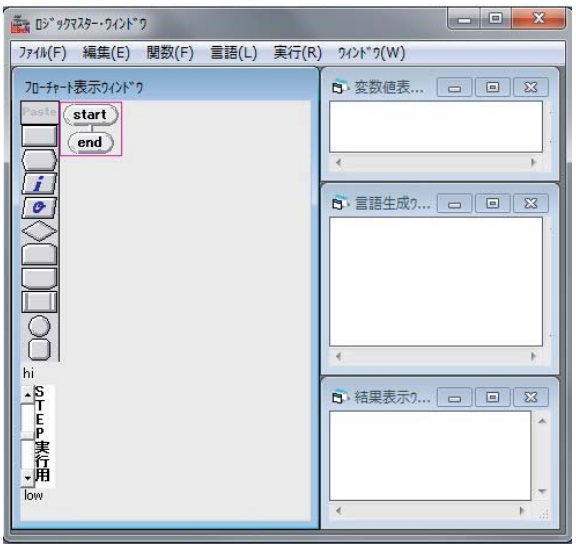
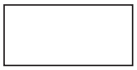
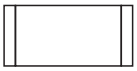
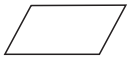
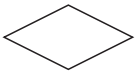
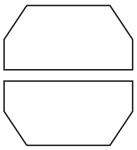


表１　流れ図記号

記号	記号名	記号の説明
	処理 (process)	任意の種類の処理を表します。よく，演算式で計算された結果の値を，変数の値として記憶するのに用いられます。
	定義済み処理 (predefined process)	関数やモジュールなど，他の場所で定義された１つ以上の演算または命令群からなる処理を表します。
	準備 (preparation)	変数の初期値設定や乱数発生の準備など，その後の動作のための最初の準備処理を表します。
	データ (data)	データ（媒体を指定しないデータ）の，キーボードなどからの入力，あるいは画面などへの出力を表します。
	判断 (decision)	１つの入口と複数の択一的な出口を持ち，記号の中の条件の評価に従って，ただ一の出口を選択することを表します。
	ループ端 (loop limit) 上図：ループ始端 下図：ループ終端	繰り返し実行の最初と最後を，ループ始端とループ終端で表します。ループ始端からループ終端に挟まれた区間の命令群は，ループ始端あるいはループ終端の中の条件の評価に従って繰り返し実行されます。ループ端の対の複合は入れ子構造になります。
	結合子 (connector)	対となる２つの同じ名前の結合子の，一方からもう一方に実行の流れが移ることで，流れ図の中の実行の流れの結合を表します。
	端子 (terminator)	流れ図の始まりと終わり，あるいは関数の入口と出口を表します。start, end, entry, return と，それぞれの記号内に表記します。

れた通りである。

図１の左端のアイコン列の一番上は後程紹介するとして，上から二番目が「処理」の記号，三番目が「準備」の記号，四番目と五番目がどちらも「データ」の記号である。

表１にあるように「データ」の記号は，入力と出力の全く逆の役割を両方共有している。

よって，同じ「データ」記号で，入力と出力の別々の役割を持つ二つの記号を区別するために，「データ」記号の中に，入力であれば筆記体英字小文字の「i」，出力であれば筆記体英字小文字の「o」を，図１の左側のアイコン列の四番目と五番目のように表示している。

六番目が「判断」の記号，七番目と八番目が「ループ端」の「ループ始端」と「ループ終端」の記号で，七番目の「ループ始端」記号は「前判定繰り返し」として繰り返し範囲の始まりに，八番目の「ループ終端」記号は「後判定繰り返し」として繰り返し範囲の終わりに，それぞれ繰り返しの条件を設定するのに使う記号である。

九番目が「定義済み処理」の記号，十番目が「結合子」の記号，十一番目が，表１には無い記号であるが，これも「結合子」の記号として扱う。

「結合子」記号は繰り返しの中から抜け出る場合に用いる記号で，十番目の「結合子」記号は強制的に「ループ端」の繰り返しから抜け出て「ループ終端」の次に移る，C 言語における補助制御文の「break」である。

十一番目の「結合子」記号は強制的に繰り返しの「ループ始端」に戻る，C 言語における補助制御文の「continue」である。

さらに，表１の一番下の記号である「端子」記号は，「フローチャート表示ウィンドウ」の中の「start」と「end」と表記した二つの「端子」記号として，図１のように初期画面の流れ図に現れ，それ以後ずっと現れ続ける。

初期画面の流れ図には，その二つの「端子」記号の他に，それらを上から下に結ぶ一本の直線の「流れ線」記号がある。「流れ線」記号も JIS で定めている流れ図記号の一つで，流れ図の実行の流れを示す線である。

「流れ線」記号は，上下の直線，あるいは左右と上下に折れ曲がった直線である。「流れ線」記号で，流れ図の中のふたつの流れ図記号を結ぶ。「流れ線」記号での実行の流れは，上から下，左から右に移っていく。

実行の流れが，逆方向の下から上，あるいは右から左に移る場合は，流れが移る方向に向かって「流れ線」記号の先端に矢印をつけることがある。しかし「logic master」では，矢印のない「流れ線」記号を用い，実行の流れを，上から下，左から右と定める。

「フローチャート表示ウィンドウ」の中の流れ図は，マゼンダ（赤紫）色の長方形で囲んで表示する。そしてその長方形の中の流れ図に対して，流れ図記号の追加挿入・移動・削除・複写などの追加編集作業を行って，目的の流れ図を作成していく。

３．アプリケーションソフトの操作（流れ図の作成）

3．1　流れ図記号の追加挿入作業

図２は，流れ図の中に「処理」記号を追加挿入する様子である。

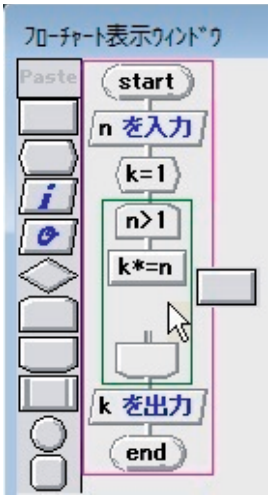
「処理」記号を流れ図の中に追加挿入するには，まず図２の左側の上から二番目の記号を左クリックする。すると，図２のようにマウスカーソルの右上に「処理」記号が現れ，マウスの動きに従って，マウスカーソルと共に移動する。

そして，流れ図の中の追加挿入したい位置の「流れ線」記号の上で，マウスカーソルを合わせると流れ図のその位置の流れ線が伸びて流れ図全体が広がり，追加挿入を受け入れる形となる。その状態で左クリックすることにより，「処理」記号を流れ図のその位置に追加挿入する。

「処理」記号を流れ図に追加挿入した後でも，マウスカーソルの右上には「処理」記号が現れたままになっているので，その状態のままさらに左クリックを続ければ，流れ図の中にいくらでも次々と「処理」記号を追加挿入することができる。

「処理」記号の流れ図への追加挿入を止めるため，一度現れたマウスカーソルの右上の「処理」記号を消すには，「フローチャート表示ウィンドウ」の中で，流れ図の範囲を示すために流れ図の周りに現れている，マゼンダ色の長方形より外側の右あるいは下の場所

図２　挿入



で左クリックする。これは，一度左側のアイコン列の中の流れ図記号を左クリックし，マウスカーソルの右上にその流れ図記号が現れるようにすると，流れ図の外側で左クリックしてその流れ図記号を消すまで，流れ図の中にいくつも追加挿入し続けることが可能であることを示す。

図２は，記号の中に「n>1」と表記した「ループ始端」記号と，記号の中は空白の「ループ終端」で上下を挟んでいる記号の中の，「k*=n」と表記した「処理」記号の下に，記号の中は空白の「処理」記号を，流れ図の中に新たにもう一つ追加挿入しようとしている様子である。

流れ図に，記号の中は空白の「処理」記号を追加挿入した後で，その「処理」記号の中に文字を入れるには，まずその「処理」記号を右クリックする。

すると，図３のような電卓の様相の「文字ウィンドウ」が現れる。そこで，「文字ウィンドウ」の中に並んである，電卓のキーのような数字や英字，あるいは「+」「−」「.」や「,」などの特殊文字の１文字からなるボタンのいずれかを順に左クリックしていく。または，パソコンのキーボードから文字を順に押していく。そうすることで，「文字ウィンドウ」内の一番上の横長の文字列表示枠の中に１文字づつ文字が入力されていくので，目的の文字列が入力し終わったら，最後に「OK」のボタンを左クリックするか，パソコンのキーボードの「Enter」キーを押す。

その結果，「文字ウィンドウ」が消え，「文字ウィンドウ」の中の文字ボタンやキーボードを使って入力した文字列が「処理」記号の中に現れる。

図３にはマウスカーソルが二つ現れているが，実際は一つである。

図３は，二つの別々のマウス操作の説明のため，二つのよく似た画面を合成して一つにした図である。

まず，二つあるマウスカーソルの左側の方は，最初のマウス操作を表していて，流れ図

図３　文字の入力

の中の「処理」の流れ図記号を右クリックしている様子である。次の右側の方は，先程の右クリックの次の動作である。その右クリックにより，画面上に「文字ウィンドウ」が現れるので，そこからが右側のマウスの操作となる。右側のマウスカーソルで，「文字ウィンドウ」の文字のボタン「n」「−」「−」を順に左クリックしている。それで，「n−−」の３文字が「文字ウィンドウ」の文字列表示枠に表示される。

そして，「OK」のボタンの左クリックか，パソコンのキーボードの「Enter」キーで文字列を決定し，「文字ウィンドウ」を消し，その「n−−」の文字列を「処理」記号の中に表示するのである。

3．2 「文字ウィンドウ」による流れ図記号への文字入力

「文字ウィンドウ」は，流れ図記号の中に文字列を表示するためのウィンドウである。このウィンドウは，文字列を入力する様子が，マウスカーソルによって成されるため，キーボードで文字列を入力するよりも，より視覚的に有効であることから用いたのである。

流れ図記号の中に表示する文字は，二行目の左端の「a」のボタンから五行目の右端の「>」ボタンまでの48文字が，そのボタンで押される順に文字列表示枠に左から右に表示されていく。

その下に，「演算子」・「数値関数」・「入力履歴」・「登録文字」の４つのプルダウンメニューがある。いずれも，流れ図記号の中に入力するのに有効なメニューである。

順に４つのメニューを紹介する。

「演算子」のメニューは流れ図を実行する場合に使える演算子のメニューで，そのメニューは「= {代入}」・「+ {加算}」・「− {減算}」・「* {乗算}」・「/ {除算}」・「% {剰余}」・

「true {論理値の真}」・「false {論理値の偽}」・「time {現時刻の秒の値}」・「< {より小さい}」・「<= {より小さいか等しい}」・「== {等しい}」・「> {より大きい}」・「>= {より大きいか等しい}」・「!= {等しくない}」・「!({～でない　論理値の否定}」・「)&&({～および～　論理積}」・「)||({～または～　論理和}」の18個からなり，それぞれを選ぶと，「=」・「+」・「−」・「*」・「/」・「%」・「true」・「false」・「time」・「<」・「<=」・「==」・「>」・「>=」・「!=」・「!(」・「)&&(」・「)||(」の文字列が文字列表示枠に表示される。なお，「{」と「}」で囲まれた部分は，文字列の説明となっているので表示されない。

「数値関数」のメニューは流れ図を実行する場合に使える数値関数のメニューで，そのメニューは，「sqr({平方根}」・「log({自然対数}」・「exp({指数}」・「sin({正弦（角度はラディアンで与える)}」・「cos({余弦（角度はラディアンで与える)}」・「atn({逆正接（値はラディアンで得る)}」・「abs({絶対値}」・「sgn({符号（負，0，正で，−1，0，1 を得る)}」・「floor({引数の値を越えない最大の整数を得る}」・「rnd({0 から 1 までの一様乱数を得る}」の10個で，それぞれを選ぶと，「sqr(」・「log(」・「exp(」・「sin(」・「cos(」・「atn(」・「abs(」・「sgn(」・「floor(」・「rnd(」の文字列が文字列表示枠に表示される。ここでも同じく，「{」と「}」で囲まれた部分は，文字列の説明となっているので表示されない。

「入力履歴」のメニューは，このソフトでは「文字ウィンドウ」で入力した文字列を，新しいものから順に20個記憶していくので，20回以内に入力していた文字列を再度入力したい場合に使えるメニューである。

「登録文字」のメニューは，

「3 章 sqr(5)*sqr(3)」・「4 章 n=1 ; n<=10 ; n++」・
「5 章 (60<=x)&&(x<=80)」・「6 章 float d[100]」・
「6 章 i%j==0」・「7 章 nen%4!=0」・
「7 章 (nen%100==0)&&(nen%400!=0)」・
「7 章 floor((nen−1)/4)」・
「8 章 saikoro=floor(rnd(1)*6+1)」・
「8 章 suuji=floor(rnd(1)*100+1)」・
「9 章 i=−5 ; i<=5 ; i++」・
「9 章 j=1 ; j<=sqr(25−i*i) ; j++」・
「9 葬 a[abs(j−k)]++」・
「10章 i=1 ; k!=a[i] ; i++」・
「10章 i=floor((l+h)/2)」・
「11章 j=i+1 ; j=>2 ; j−−」・
「11章 j=i+haba ; j=>1+haba ; j−=haba」・「11章 l=floor(n/2) ; l>=1 ; l−−」・
「11章 p=a[(s+e)/2]」・「11章 i<e」の20個で，それぞれを選ぶと，「sqr(5)*sqr(3)」・「n=1 ; n<=10 ; n++」・「(60<=x)&&(x<=80)」・「float d[100]」・「i%j==0」・「nen%4!=0」・「(nen%100==0)&&(nen%400!=0)」・「floor((nen−1)/4)」・「saikoro=floor(rnd(1)*6+1)」・

図４　「登録文字」のメニュー

「suuji=floor(rnd(1)＊100+1)」・「i=−5 ; i<=5 ; i++」・「j=1 ; j<=sqr(25−i＊i) ; j++」・「a[abs(j−k)]++」・「i=1 ; k ! =a[i] ; i++」・「i=floor((l+h)／2)」・「j=i+1 ; j=>2 ; j−−」・「j=i+haba ; j=>1+haba ; j−=haba」・「l=floor(n／2) ; l>=1 ; l−−」・「p=a[(s+e)／2]」・「i<e」の文字列が文字列表示枠に表示される。章番号は，文字列の説明となっているので表示されない。

このソフトを使ったアルゴリズムの授業では，春学期と秋学期でそれぞれ15回の年間総合計30回開講しているのであるが，そこでは０章はゲームを配布，１～３章は PDF 教材を配布，４～11章はネット公開による e−learning 教材を提示して授業を行っている。

それぞれの章立ての内容は，

０章　アルゴリズムゲーム
１章　流れ図と流れ図記号
２章　流れ図の構造化プログラミングの基本構造の３つ
３章　計算式（四則演算と平方根），文字の表示と入力，平方根などの関数，論理演算式，前・後判定繰り返しと繰り返し中止
４章　各種の数列表示とコンピュータ誤差
５章　得点の合計・平均・最大・最小を求める
６章　約数の数・最大公約数・最小公倍数・素数・コラッツの問題，および関数定義
７章　日数計算（うるう年・曜日計算・カレンダー表示）
８章　コンピュータの乱数ゲーム（サイコロ・数当ておよびバイナリーツリーによる数

の予測）
9章　文字記号によるグラフ表示（棒グラフによる様々な図形と乱数による表示）
10章　サーチ（シーケンシャルサーチ・番兵法とバイナリサーチ）
11章　ソート（セレクション・バブル・シェーカー・インサート・シェル・ヒープ・クイック）

となっていて，そこでよく使用する命令文を，授業の補助となるように「登録文字」のメニューの中で用意している。

図4は，7章の内容のうるう年計算に必要な，判断命令の中の条件を表した文字列を選んでいる様子である。

3．3　流れ図記号の移動・削除・複写作業

流れ図の中の流れ図記号を別の場所に移動するには，まず流れ図の中で移動したい流れ図記号を左クリックする。

例えば図3の流れ図で，「n>1」と表記された「ループ始端」記号と空白の「ループ終端」で囲まれた部分を流れ図の一番下，すなわち「end」と表記した「端子」記号のすぐ上に移動するには，まず「n>1」と表記された「ループ始端」記号を左クリックする。

すると，図5のようにマウスカーソルの右上に「BLOCK」の文字の四角い形が現れ，「n>1」と表記された「ループ始端」記号と空白の「ループ終端」で囲まれた部分が流れ図の中から消える。その消えた部分は，「BLOCK」の文字の四角い形として「logic master」が記憶する。

また，後程紹介するとしていた「フローチャート表示ウィンドウ」の中の左側の縦に一列に並んでいる一番上のアイコンに，以前は薄っすらと現れていた「Paste」の文字が，今度は図5のようにはっきりと現れる。このアイコンを「Paste」アイコンと呼ぶ。

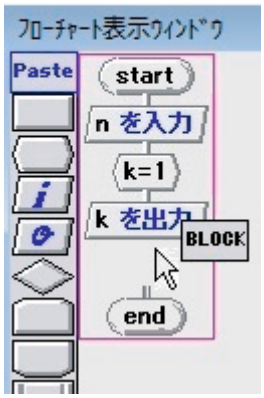
「logic master」では，流れ図の中の流れ図記号を左クリックすると，流れ図の中からその流れ図記号が消え，マウスカーソルの右上に「BLOCK」の文字の四角い形が現れると同時に，消えた流れ図記号を「BLOCK」の四角と同じく「Paste」アイコンにも記憶するのである。

「フローチャート表示ウィンドウ」の左の一番上のアイコンに「Paste」の文字がはっきりと現れている場合は，流れ図の中の最後に左クリックして消した流れ図記号を記憶している。

「BLOCK」の四角がマウスカーソルの右上に現れている状態で，流れ図の中の移動先の「流れ線」記号の上にその状態でマウスカーソルを合わせると，流れ図のその場所が広がり，移動を受け入れる形となる。そこで左クリックし，「n>1」と表記した「ループ始端」記号と空白の「ループ終端」で囲んだ部分を移動する。

一度現れたマウスカーソルの右上の「BLOCK」の四角を消すには，マウスカーソルの右上に現れた流れ図記号を消す場合と同様，「フローチャート表示ウィンドウ」の中で，

図５　移動・削除・複写



流れ図の範囲を示すマゼンダ色の長方形より外側の右あるいは下の位置で左クリックする。

流れ図の中の流れ図記号を削除するには，移動と同様，まず削除したい流れ図記号を左クリックする。すると図５のように，マウスカーソルの右上に「BLOCK」の四角が現れ，左クリックした流れ図記号が流れ図の中から消える。そして，流れ図の範囲を示すマゼンダ色の長方形より外側の右あるいは下の位置で左クリックして，マウスカーソルの右上の「BLOCK」の四角を消す。

流れ図の中の流れ図記号をいくつも複写するには，移動と同様，まず複写したい流れ図記号を左クリックする。するとこれも図５のようになる。その後，マウスカーソルの右上に「BLOCK」の四角が現れている状態で，元の流れ図の場所と，その他複写したい流れ図の場所で左クリックする。

流れ図記号をいくつも複写するのに「Paste」アイコンを使う方法もある。

「Paste」アイコンに「Paste」の文字がはっきりと現れている状態で，「Paste」アイコンを左クリックする。するとマウスカーソルの右上に「Paste」の文字の四角い形が現れる。後は，マウスカーソルの右上に「BLOCK」の文字の四角い形が現れている場合と同様である。

４．アプリケーションソフトの操作（流れ図の実行）

4．1　流れ図記号の実行

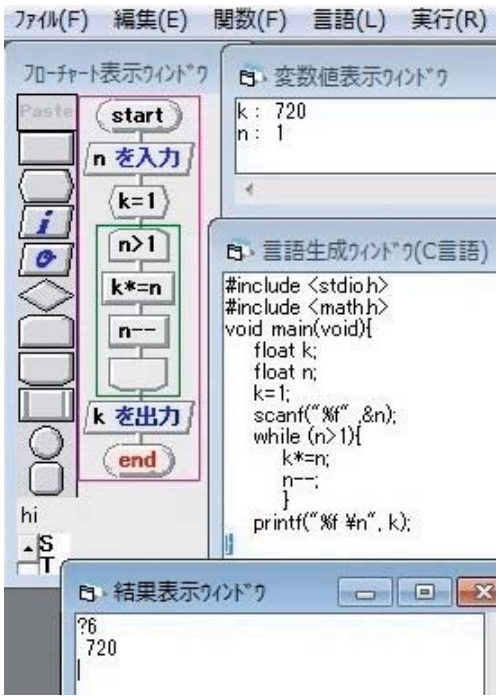
図３の作成中の流れ図を完成させ，その流れ図を実行した様子が図６である。これは階乗の値を求める計算の流れ図で，６の階乗である６！を計算し，その値である720を表示している。

では，図６の流れ図の中の流れ図記号を上から順に説明していく。

まず，「start」と表記している「端子」記号に続く入力用「データ」記号には「nを入力」と表記している。

これは「データ」記号が入力用であることを示すため，この論文では白黒印刷で色が判

図６　階乗計算の実行



別できないが，変数名の「n」の後ろに青色で「を入力」の文字が表示されている。

「logic master」では，流れ図の中の流れ図記号に表記する文字は黒色である。ただし，「データ」記号での表示については，入力用と出力用が区別できるように，それぞれ「を入力」と「を出力」の文字を青色で，「logic master」により追加表示するようにしている。

その下の「準備」記号には「k=1」と表記している。それは，これから階乗の値を計算していく準備として，変数 k に初期値1を代入している記号である。

さらにその下の流れ図記号は，「n>1」と表記した「ループ始端」記号と空白の「ループ終端」で囲まれた，繰り返しの部分である。この繰り返しの部分は，変数 n の値が 1 よりも大きい間，繰り返して実行する事を示している。

具体的には，変数 n の値が 2 以上の間，「ループ端」で挟まれた「処理」記号の「k*=n」と「n−−」を何度も繰り返して実行する。

図 6 の流れ図を実行すると，変数 k には最初に「文字ウィンドウ」で入力した値が変数 n に入る。すなわち変数 n に値 6 が入り，それに続く 2 つの「処理」記号の繰り返しで，「k*=n」と「n−−」により，変数 k の値と変数 n の値が，6*1 の 6 と 6−−の 5，6*5 の30と 5−−の 4，30*4 の120と 4−−の 3，120*3 の360と 3−−の 2，360*2 の720と 2−−の 1 と変化し，変数 n の値が 1 より大きくなくなったので，その繰り返しの 5 回目で終了する。

そして実行の流れが，次の出力用「データ」記号に移る。

出力用「データ」記号では，「k」を表記しており，これも入力用の場合と同様，出力用であることを示すため，白黒印刷で色が判別できないが，その文字に続いて青色で「を出力」の文字を，「logic master」により追加表示している。この出力用「データ」記号を実行して，変数 k の値の720を「結果表示ウィンドウ」に出力する。

図６では「フローチャート表示ウィンドウ」の他に，「変数数値表示ウィンドウ」・「言語生成ウィンドウ（C 言語）」・「結果表示ウィンドウ」を，各ウィンドウの中の文字がすべて見えるようにして表わしている。

「変数数値表示ウィンドウ」では，流れ図で使用した変数 k と n の実行終了後の，それぞれの値720と 1 を表示している。

また「言語生成ウィンドウ（C 言語）」では，流れ図の表現を，C 言語で表した文字列に置き換えたものを表示している。

そして「結果表示ウィンドウ」では，入力用「データ」記号の実行時に，入力を促すためのプロンプトとして常に表示する「?」の文字と，変数 n に入力した値の 6，さらに出力用「データ」記号の実行時の変数 k の値の720を表示している。

4．2　再帰関数の実行

図６の実行例で紹介した階乗の計算は，「logic master」で用意した再帰関数で実行することができる。関数は，図１で説明のあった上から九番目の「定義済み処理」の記号を用いる。関数を使うには，まず流れ図の中に「定義済み処理」の記号を挿入する。そしてその記号の中に，図３のようにして「文字ウィンドウ」で関数名 kaijo を入力する。

次にメニューの「関数」の「同一関数内表示上限（1)」を選んで，図７の左側の流れ図表示から右側の流れ図表示に変える。そして，関数 kaijo の中を図８の左のように作り上げる。その結果，繰り返し命令はないのであるが，変数 n の値が６の場合は再帰関数により，実行は，図６の繰り返しの中身と同様，図８の右のような流れになり，６の階乗として変数 k の値の720を「結果表示ウィンドウ」に出力する。

図７　関数 kaij の定義

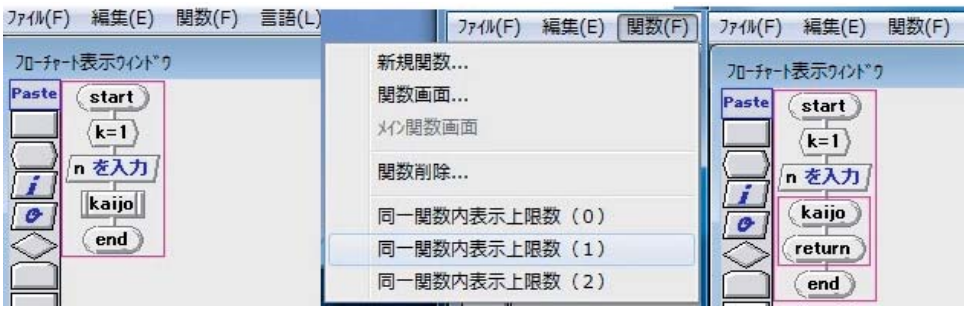
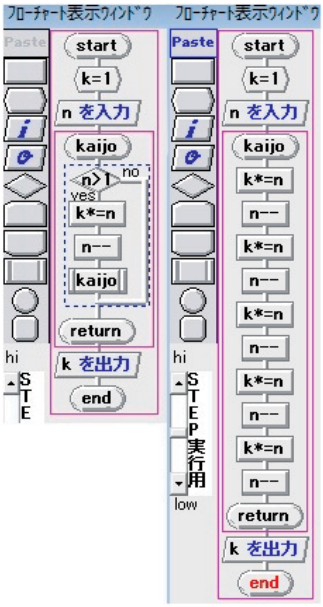


図８　再帰関数による階乗計算の流れ図



5．お　わ　り　に

本アプリケーションソフトによるアルゴリズム学習の効果を調査するため，図９のようなアンケートをアルゴリズム論の受講生に配布・回収し，それらの回答を得た。

その結果，図10のように，全体90人中，プログラミング言語も流れ図も使ったことのない人は59人で，どちらか一方または両方使った人は31であり，それぞれの平均は，すべて

図９　アンケート用紙

この授業を受ける前に，プログラミング言語を使ったことがあるか？

はい　　いいえ

はいの人は言語の名称を。

C 言語　JAVA 言語　C++　その他（　　）

この授業を受ける前に，流れ図を使ったことがあるか？

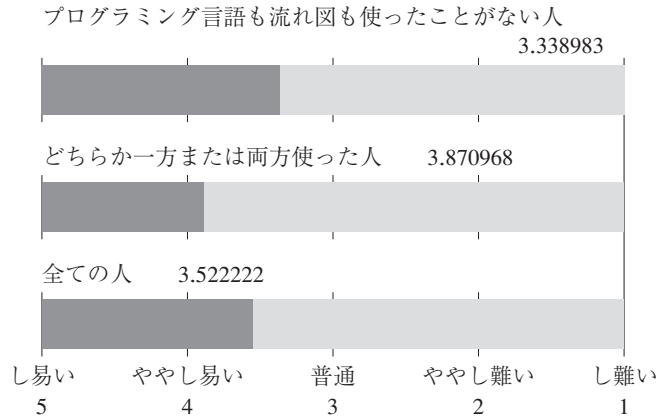
はい　　いいえ

流れ図のソフトについて
この授業のような，プログラミング言語を使わず
流れ図のソフトによるアルゴリズムの学習について，
学習し易いかどうか次の五段階から選び，
なぜそう思うかを述べよ。

し易い　　やや し易い　　普通　　やや し難い　　し難い

理由：

図10　アンケート結果



「ややし易い」と「普通」の中間にあった。

そして，プログラミング言語も流れ図も使ったことのない人は，どちらか一方または両方使った人よりも「普通」に近い結果となっている。

プログラミング言語も流れ図も使ったことのない人の回答理由の主なものは，「この種の学習が初めてのために，他と比べようがないが分かりやすいのではないかと思った」であり，どちらか一方または両方使った人では，「プログラミング言語の書き方や文法によらないから分かり易い」であった。

これからも，本アプリケーションソフトの発展が有用であると結論付けた。

参　考　文　献

[1] 太田幸一：「アルゴリズム学習用ソフトの開発」，教育システム情報学会研究報告 p. 158-161（2007-3）

[2] 太田幸一：「プログラミングゲーム」，ゲーム学会第５回合同研究会研究報告 p. 3-4（2007-7）

[3] 太田幸一：「アルゴリズム学習の教材」，平成19年度大学教育・情報戦略大会　私立大学情報教育協会 p. 114-115（2007-9）

[4] 太田幸一：「プログラミング学習のためのパズルゲーム」，ゲーム学会「ゲームと教育」研究部会研究会報告 p. 5-7（2013-3）