構造化チャートを用いたアルゴリズム学習支援システム

斐品正照[†] 徳岡健一[††] 河村一樹[†]

プログラミング言語を用いたアルゴリズム学習は，高校生や非理工科系の大学生にとって難しいことがある．そこで，筆者らは JPADet（Japanese PAD editor and interpreter）と呼ぶ学習支援システムを設計・開発した．JPADet は，プログラミング言語に依存しないで，日本語と構造化チャート（PAD: Problem Analysis Diagram）を用いた図解によるアルゴリズムの作成と実行が可能である．これらを評価・検証するために，筆者らは実際の授業においてこのシステムを試用し，プログラミング言語を用いた場合と，JPADet を用いた場合の学習効果を比較した．その結果，JPADet を用いた場合の学習の方が効果的であったことが示された．

Algorithm Learning Support System with Structured Chart

Masateru Hishina,[†] Ken-ichi Tokuoka[††] and Kazuki Kawamura[†]

Learning algorithms through programming language is difficult for high-school students as well as for non-science/engineering undergraduate students. In order to overcome such difficulties, we have designed and developed a learning support system, named “JPADet (Japanese PAD editor and interpreter)”, characterized by supporting the algorithm implementation based on Japanese words together with the structure chart PAD (Problem Analysis Diagram). To show the effectiveness of our proposal, we have used our prototype system in real programming classes. We have compared the learning efficiency of “learning through programming language” versus “learning through JPADet”. The results show that our JPADet system is much better when compared with learning through programming language, to support students learning algorithms.

1. はじめに

情報処理学会の一般情報処理教育部会が策定した，情報処理を専門としない利用者の立場が主となる大学生に対するプログラミング教育に関する調査研究報告書がある[1]．この報告書では，プログラミング教育に関して,「プログラミング」教育（かぎ括弧付き）とプログラミング教育（かぎ括弧なし）とに分けている．前者は，特定の実用言語の習得だけを目的とせずに，コンピュータサイエンスの基本的な概念を理解するための教育である．後者は，実用言語の習得を目的としたプログラマ養成のための教育である．一般情報処理教育におけるプログラミング教育は，前者を目指すべきと提言されている．このことから，一般情報処理教育におけるプログラミング教育については，実用言語に依存することなく，コンピュータサイエンスの基本的な概念を理解させるための方策が必要になってくる．

一方，2003 年度から，高等学校普通科において教科「情報」が新設された[2]．その中の科目「情報 B」では，モデル化とシミュレーションという単元でプログラミングが扱われることになる．ここでも,「プログラミング言語の習得が目的とならないように」という留意点が明示されている．これより，上述した大学の一般情報処理教育における「プログラミング」教育とほぼ同じような状況におかれることになる．

しかし，プログラミング言語を用いた大学の一般情報処理教育では，アルゴリズムの理解よりもプログラミング言語の構文に学習者の興味・関心・注意が偏ってしまうという傾向が報告されている[3]．これらの原因の 1 つとして，プログラミング言語が構文中心の文的表象であり，1 次元的にアルゴリズムを表現する必要があることが考えられる．

このようなことを解決するために，筆者らは大学での一般情報処理教育，および高等学校での教科情報を

† 東京国際大学商学部
Department Information Systems, School of Business & Commerce, Tokyo International University
†† 株式会社日本科学技術研修所
The Institute of Japanese Union of Scientists & Engineers, Inc.

視野に入れながら，プログラミング言語にまったく依存せずに，かつ2次元的に図（一般的な規格に準ずるシンボルの表現）を用いてアルゴリズム学習が可能になる学習支援システムについての研究を行っている．

本論文では，筆者らが開発したアルゴリズム学習支援システムの概要について述べてから，実証実験におけるプログラミング言語と構造化チャートを用いた場合の学習の比較検証について述べる．具体的には，2章で筆者らの研究の位置づけ，3章で設計・開発したシステムの概要，4章で実証実験の概要，5章で考察について述べる．

2. 研究の位置づけ

2.1 図を用いたプログラミング教育

2次元的に図的表象としてアルゴリズムを表現する方法はすでにいくつか考えられており，具体的には流れ図（フローチャート）や構造化チャートなどを用いることが一般的である．しかし，よく用いられる流れ図にはいくつかの問題があることがすでに指摘されている[4]．その1つとしては，矢印を多用することにより，プログラムの制御構造が不明確になることがあげられている．このため，構造化プログラミングの表現に適した構造化チャートが提案されており，国際規格であるISOをはじめ日本工業規格であるJISでも規定されている．JISでは，PSD，DSD，SPD，HCP，PAD，LCPをプログラム構成要素の表記法として規定している[5]．

このうち，PAD（Problem Analysis Diagram）は1970年代後半に，川合敏雄と二村良彦により考案されたものであり，ワーニエ法の表記法を図式化して拡張したものである[6),7)]．流れ図では，矢印と流れ線により上下左下と4方向にアルゴリズムが展開できることから，アルゴリズム全体の制御構造が複雑になりやすいといえる．しかし，PADは，処理の流れである基本線を上から下に，アルゴリズムの構造を左から右に，合計2方向だけのシンプルなアルゴリズムの展開が可能になる．

このような図を用いてプログラミングの教育に活用する研究・報告事例はいくつかある．図に用いられるシンボルの表現が一般的な規格に準じていないようなもの[8)]や，流れ図を用いるもの[9)]がある一方で，構造化チャートを用いるものも存在する[3),10),11)]．これらの教育活動のほとんどは，作成した図を最終的にはアセンブリ言語やCOBOL，C言語，Prologになどに変換する過程があり，何らかの形でプログラミング言語に依存しているものが多い[3),9)~11)]．

2.2 図を用いたアルゴリズム学習支援システム

筆者らは，JPADet（Japanese PAD editor and interpreter）と呼ぶ学習支援システムを設計・開発した．JPADetは，プログラミング言語にまったく依存することなく，日本語と構造化チャートの1つであるPADを用いた図解によってアルゴリズムの作成と実行を，1つのシステム上において可能にしたものである．なお，JPADetはスタンドアロン環境におけるシステムの提供（具体的にはCD-ROM）を想定している．このシステムの概要については3章で述べることにする．

筆者らのJPADetのように，PADを用いた学習支援システムはいくつかあり，比較的近いものとして，遠矢らのFOXPADや，石田らのPADエディタを用いたアルゴリズム学習支援システムがあげられる[11),12)]．筆者らのJPADetと比較すると，いずれも図に用いられるシンボルの表現が一般的な規格に準じていることは共通している．遠矢らのFOXPADに関しては，スタンドアロン環境を想定していることはJPADetと共通している．しかし，FOXPADによる学習支援は，作成したPAD図に基づくC言語のソースコードの骨組み（スケルトン）を生成（さらにコードの加筆・修正などの記述が必要）するまでであり，アルゴリズムの実行確認がC言語のコンパイラに依存している[12)]．一方，石田らのPADエディタを用いたアルゴリズム学習支援システムに関しては，プラットフォームに依存しないWWW（World Wide Web）での提供を目指してJavaアプレットによるシステムの構築を行っており，スタンドアロン環境を想定しているFOXPADやJPADetとは異なる．また，作成したPAD図とC言語のソースコードの両方向の変換が可能である点はFOXPADやJPADetとも異なるが，アルゴリズムの実行確認自体は，現段階では，C言語のコンパイラに依存しているようである[11)]．

筆者らの研究において開発したJPADetはPAD図の作成だけでなく，そのアルゴリズムの実行確認自体も1つのシステム上で可能な点が特徴であり，「プログラミング」教育を想定したことによりプログラミング言語に依存していない（そのソースコード自体は隠蔽している）点も特徴である．JPADetも石田らのシステムを見習い，将来的にはプラットフォームに依存しないWWW環境での提供を行うつもりだが，筆者らの研究では，大学での一般情報処理教育と，高等学校での教科情報の両方を視野に入れており，高等学校のインターネット接続環境が進展するまで（生徒ひとり1台のインターネット接続環境が実現するまで）は，CD-ROMなどの媒体によるスタンドアロン環境が妥

当であると判断した．

3. アルゴリズム学習支援システムの概要

3.1 基本仕様

(1) 前提条件

本研究では，一般情報処理教育段階の大学生あるいは高校生を対象にして，アルゴリズムの習得，論理的思考力および抽象的概念の理解力の育成を目指している．このため，JPADet の設計にあたっては，次のような前提条件を設定した．

(a) 実務用ではなく教育用を前提にすること
(b) 大学の一般情報処理教育課程（1・2 年生）におけるプログラミング演習科目や高等学校教科目「情報 B」での演習に適用することを想定すること
(c) 実用言語に依存することなく，アルゴリズムの設計が学習できること
(d) 構造化チャート（PAD）を用いることで，アルゴリズムを視覚的に記述できること
(e) 日本語を使用して簡単な式表現ができること
(f) トランスレータではなく，インタプリタとして実装すること

このうちの (f) のインタプリタ採用の理由として，開発工数の軽減，インストール時の設定が単純，トランスレータのようなコンパイラ製品ごとの差を考慮する必要がない，利用者にとって言語の知識はまったく不要，将来的に学習解析機能などを独自に追加可能，などがあげられる．

(2) JPADet の言語仕様

ここでは，JPADet の箱内記述の言語仕様についてまとめる．

① データ型

数値型（整数と浮動小数点の区別は利用者に対して隠蔽）と文字型を用意する．

② メインと副プログラム

JPADet 上の手続きは，プログラムファイルにただ 1 つ存在するメインと，関数もしくは手続きとして実行される副プログラムから構成される．副プログラムは，式あるいは文が書ける箇所で，

副プログラム名（実引数，…）

と書くことで呼び出せる．実引数と仮引数の数は一致していなければならない．副プログラムを関数として用いる場合は，単一の戻り値を持つ．引数は値渡しのみで参照渡し（関数内の引数への代入が呼び出し側の実引数変数へ戻ること）はしない．

③ 制御構造

制御構造はすべて PAD（処理箱，双・多岐選択箱，前判定繰返し箱，後判定繰返し箱，条件値判定繰返し箱，入力箱，出力箱）で記述されるので，箱内記述は制御構造を持たない．なお，goto 文は扱わない．

④ 変数

使用する変数は，専用の宣言ダイアログで宣言する．メインで宣言された変数は，プログラム全域に対応するスコープを持つ．副プログラムで宣言された変数には，仮引数（呼び出しプログラムから引数として受け取る変数），一時変数（副プログラム内でのみ有効な変数），「結果」（値を返す副プログラムでの結果を代入する変数）がある．ただし，関数が終了した時点で「結果」に設定された値が戻り値になる．このため，明示的な return 箱は持たない．

⑤ 配列

変数は配列として使用することができる．配列変数は，

変数名［添字，…］

の形で参照および代入する．

⑥ その他のデータ構造

ポインタと構造体については，複雑になるので扱わない．ファイルに関しては，シーケンシャルアクセスだけにし，テキスト関数で処理を記述する．

(3) JPADet の文法定義

構文解析にともなう構文規則は，付録 1 のように定義する．

3.2 実装環境

(1) 動作環境

JPADet が動作する OS は，Windows95/98/Me および/WindowsNT4/2000/XP とする．

(2) 開発環境

JPADet の開発には，Microsoft Visual C++6.0 およびクラスライブラリとして MFC（Microsoft Foundation Class）を用いる．また，JPADet は，MDI（Multiple Document Interface）アプリケーションとして開発する．

(3) 編集系

図 1 に示すように，編集系は CJPadetDoc（jpd ファイルの内部表現）を中心に，定義された関数・手続き（CProcedure），関数・手続き内の変数（CVar），PAD シンボル（CAlg）を管理し，要求に応じてこれらを編集する．

① CJPadetDoc（ドキュメント）

下位の構成要素である関数群を管理する．また，MFC フレームワークによるシリアライズにより，ファイルのロードとセーブも行う．関数ダイアログ

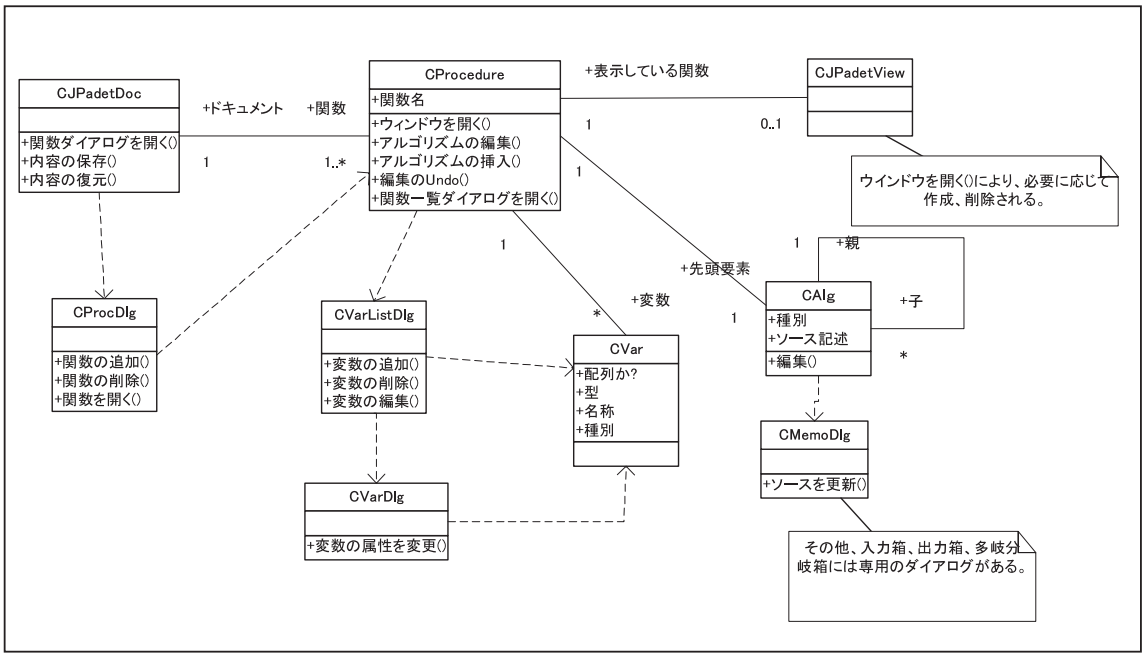


図 1 編集系の UML
Fig. 1 UML of the editing system.

（CProcDlg）を経由して，関数の追加，削除，オープン処理を行う．

② CProcedure（関数，手続き）

下位要素として，変数と処理シンボルを持つ．関数ダイアログからの要求により，対応する CJPadetView と結びついて画面上に描画される．また，結びついている（自身を表示している）ビューからの要求により，処理シンボルの木構造を編集する．変数一覧ダイアログを経由して，関数中の変数リストを編集する．

③ CAlg（処理シンボル）

属性として，種別とソース記述（任意のテキスト）を持つ．種別は，次のとおりである．

```
enum alg_code {
  unit_alg,      先頭要素
  process_alg,   処理箱
  for_alg,   条件値判定繰返し
  while_alg,     前判定繰返し
  repeat_alg,    後判定繰返し
  switch_alg,    分岐処理
  case_alg,      分岐処理中の個々の分岐
  input_alg,     入力箱
  output_alg     出力箱
};
```

任意個の子を持つことができ，unit_alg を先頭とする木構造を構成する．unit，for，while，repeat，case は，unit，case 以外の子を持つことができる．switch は，case のみを子として持つことができる．

④ CVar（関数，手続きごとの変数）

属性として，名称，配列型か否かのフラグ，型（数値か文字列か），種別（大域変数，ローカル変数，引数，結果変数）を持つ．

(4) 表示系

図 2 に示すように，表示系は，CJPadetView（MDI の関数表示画面に相当）が中心になる．CJPadetView は画面上の表示要素オブジェクトを管理し，利用者からのマウス操作やメニュー操作を翻訳し，CProcedure を経て編集操作を支援する．

① CJPadetView（MDI の関数表示画面に対応）

表示している関数（CProcedure）と 1 対 1 に対応していて，CProcedure 下のアルゴリズム木から画面上の表示要素を作成して維持する．表示要素群は，CProcedure が編集されるたびに再編成される．また，利用者のマウス操作やキーボード操作によって，シンボルの選択動作を管理し，編集指示に応じて CProcedure へ編集を要求する．

② CFigure（画面上の表示要素の基底クラス）

画面上での表示領域の座標を維持する．また，要求に応じて，画面上に自分自身を描画するメソッドと選択表示を行うメソッドがある．

③ CLine（画面上の横線）

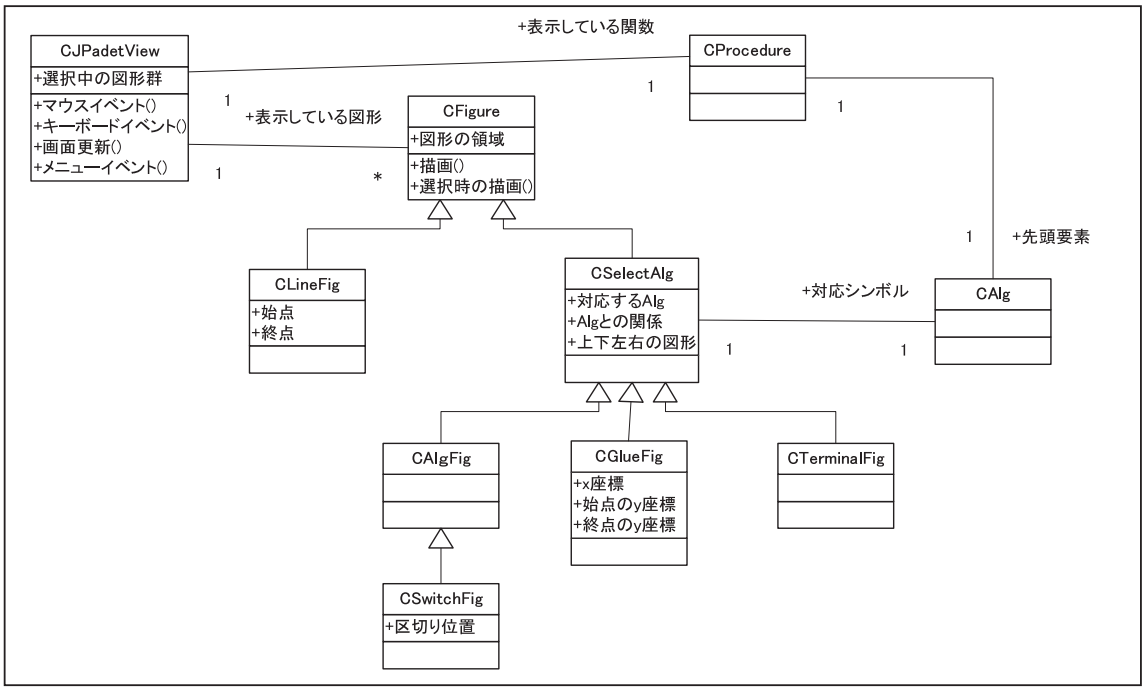


図 2 表示系の UML
Fig. 2 UML of the display system.

たんに表示されるだけで，選択はできない．

④ CSelectFig（画面上の選択可能な要素）

CAlg への参照と，その CAlg との相対的な関係を維持している．関係はシンボル自身（CAlgFig），シンボルの下方（CTerminalAlg，CGlueAlg），シンボルの子の先頭（CTerminalAlg）がある．これにより，どのシンボルがどのように選択されているかが決定されるので，適切な編集を行うことができる．

⑤ CAlgFig（多岐選択箱以外の箱）

参照している CAlg により表示形式が異なる．箱の内部には，CAlg のソーステキストがそのまま表示される．

⑥ CSwitchAlg（多岐選択箱）

多岐分岐の区切りの位置を維持していて，多岐選択箱を描画する．表示されるソースも対応している switch_alg ではなく，下位の case_alg 群のソースを表示する．

⑦ CGlueFig（画面上の縦線）

⑧ CTerminalFig（シンボルの上下端の選択領域に対応）

選択表示のときには反転されるが，通常では表示されず，位置情報だけが維持される．

(5) 実行系

実行系は，CContext が中心になる．CContext が 1 つの関数を実行するための環境（変数テーブル，現在実行中の関数（(CProcedure)，シンボル（CAlg））を維持している．関数呼び出しが行われると，新たにコンテキストが作成される．

① 字句解析

字句解析は，1 文字先読みで処理する．その際に，マルチバイト文字は，対応するシングルバイト文字があればシングルバイト文字に変換して処理する．たとえば，× は * へ，÷ は/へ，≠ は <> へ，≤ は <= へ，≥ は >= へ，それぞれ変換する．字句としては，識別子（ident），文字，文字列（string），数値（numeric）とする．

② 構文解析

上述した構文規則に従い，単純な再起下降型パーサで処理する．

③ 実行

実行は，CProcedure 上の CAlg を先頭から解析結果に応じて tree-walk することで行う．CAlg のソーステキスト解析と同時にコンテキストの変数参照を行い，同時に式評価も行うことから，中間言語は生成しない．

3.3 操作手順

ここでは，JPADet の操作手順を通して，JPADet のシステム構成や表示デザインについて取り上げる．

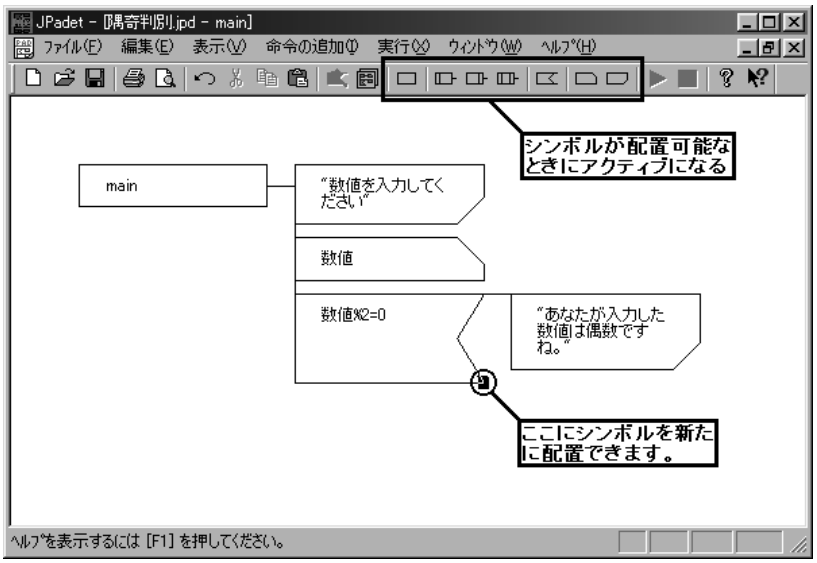


図 3 JPADet のインタフェースとシンボル位置の指定
Fig. 3 An example of the interface and symbol establishment in JPADet.

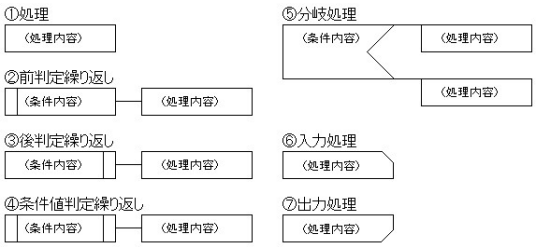


図 4 7 つのシンボル
Fig. 4 Seven symbols in JPADet.

(1) 起動

JPADet の実行ファイルへのショートカットアイコンをダブルクリックする.

(2) 初期画面の表示

JPADet が起動すると，**図 3** のような，タイトルバー，メニューバー，ツールバー，シートウィンドウ(main の箱のみ表示)，ステータスバーから構成される初期画面が表示される.

このうちのツールバーは，通常よく使われるメニューコマンドとシンボルを配置するコマンドをアイコンとして登録している.

これより，JPADet では**図 4** のような 7 つの PAD シンボルが使用できる．具体的には，処理箱，前判定繰返し箱，後判定繰返し箱，条件値判定繰返し箱，分岐処理箱，入力箱，出力箱が利用可能である．これらをシート上に配置し，必要な変数や配列を設定してアルゴリズムを作成していくことになる.

(3) 変数の作成

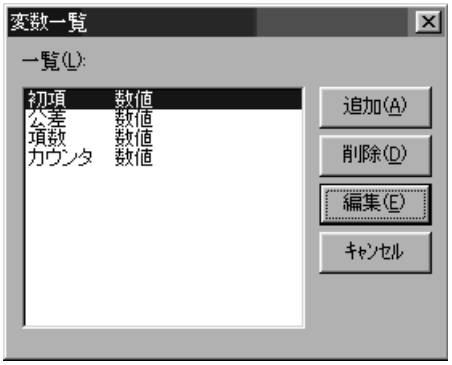


図 5 変数一覧ダイアログ
Fig. 5 Variable look dialogue.

使用する変数は，あらかじめ**図 5** のような変数一覧ダイアログを用いて宣言する.「追加」を選ぶと,［変数の編集］ダイアログが表示され，ここで変数を作成する．変数名には，日本語を指定できる．変数の型には，数値（整数か浮動小数点）と文字［列］（単一文字か文字列）と配列（添字の開始番号は 0 から）を指定できる.

(4) PAD シンボルの記入

シンボルを配置するには，シート上のシンボルを配置したい位置にカーソルを移動し，シンボルバーの中から配置したいシンボルのアイコンをクリックする．または,［命令の追加］メニューから配置したいシンボルを選択する.

① 位置の指定

シンボルを配置したい位置は，マウスまたはキー

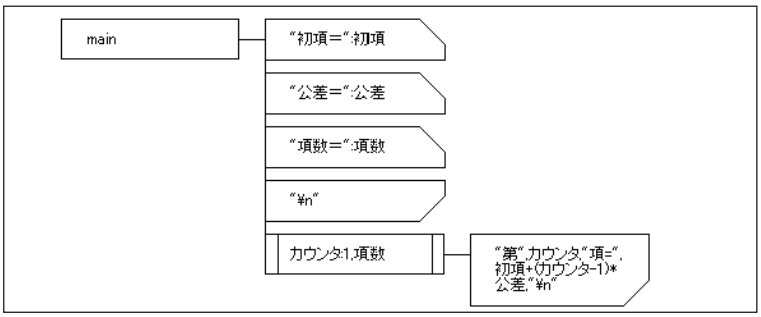


図 6 メインシート
Fig. 6 Main sheet of the sample.

ボードの方向キーで選択できる．実際にシンボルを配置する箇所は，短形（■）で表示される．シンボルが配置可能な場合は，ツールバーの各シンボルボタン（図 3 の四角で囲んだ部分），および，［命令の追加］メニューの各シンボルがアクティブになる．

シンボルの上下へ挿入する場合は，左隅やシンボルをつなぐ線をクリックする．各［繰返し］シンボルや［分岐］シンボルで，アルゴリズムを右に展開する場合は，シンボルの右端をクリックする．

② シンボルの配置

シンボルが配置可能な状態で，［命令の追加］メニューから配置したいシンボルを選択すると，短形で表示されていた箇所にシンボルが配置される．

(5) PAD シンボルの編集

PAD シンボルの編集には，以下のようにそれぞれ該当するダイアログが表示され，その中に記述する．

- ［処理］シンボル：［式］ダイアログ（代入式や計算式を記入）
- ［前/後判定繰返し］シンボル：［式］ダイアログ（繰返し条件と，条件が真のときの処理を記入）
- ［条件値繰返し］シンボル：［式］ダイアログ（繰返し制御変数名：開始値，終了値，増減分値，および，条件は真のときの処理をそれぞれ記入）
- ［分岐］シンボル：［分岐］ダイアログ（省略時は分岐が 1 個だが，多岐選択のため分岐を増やすことも可能．選択条件を記入）
- ［出力］シンボル：［出力］ダイアログ（画面かファイルかを選択したうえで，出力したい変数や式を記入）
- ［入力］シンボル：［入力］ダイアログ（キーボードかファイルかを選択したうえで，入力する変数を記入）

(6) プログラムの実行

作成したプログラムの実行を行う場合は，［実行］メニューの［実行］を選択する．また，ツールバーの［実行］ボタンをクリックしてもよい．これによって，実行ウィンドウが開き，プログラムが実行される．

また，プログラムを実行した際に，プログラムに記述エラーが含まれている場合は，エラーメッセージが表示され，エラー箇所のあるシートを開きシンボルを選択状態にする．その後，エラーメッセージに従い，プログラムのエラー箇所を修正する．

(7) プログラムの印刷

プログラムの印刷については，印刷するシートのウィンドウをアクティブにして，［ファイル］メニューの［印刷 ...］を選択する．これにより，［印刷］ダイアログが表示されるので，各項目を設定し，［OK］ボタンをクリックする．なお，印刷プレビューも可能である．

(8) プログラムの保存

プログラムの保存については，［ファイル］メニューの［名前を付けて保存 ...］を選択し，［ファイル名］ダイアログボックスにシートの名前を入力し，［保存］ボタンをクリックする．

3.4 サンプルプログラムの例

ここでは，等差数列のサンプルプログラムを JPADet で作成した例を示す．

(1) プログラム課題

等差数列の初項（a），公差（d），項数（n）を入力して，初項から第 n 項（n は項数）までの各項の値（a_n）を求める．具体的には，$a_n = a + (n-1) * d$ という関係になる．

(2) 変数一覧ダイアログ

本サンプルプログラムの変数一覧ダイアログは，図 5 のようになる．

(3) JPADet のメインシート

変数一覧をもとに，実際の PAD は**図 6** のようになり，これをメインシート上に配置する．このように，main 箱の右側に，PAD シンボルを，上から下へ（順次処理），左から右へ（選択処理か繰返し処理），展開

していくことになる．

(4) 実行結果

［実行］メニューから実行すると，実行ウィンドウが開き，初項の問合せとなる．初項・公差・項数をキーボードから入力すると，**図 7** のような結果が表示される．

4. 実証実験の概要

筆者らは JPADet を評価するための実証実験を行った．

JPADet の想定する学習者は，一般情報処理教育を受ける大学生，および教科情報の教育を受ける高校生である．今回，実証実験の対象としたのは文科系の大学生であり，筆者の中の 2 名がそれぞれ担当する演習科目を受講した 1 年生である．この 2 つの演習を実証実験の対象にすることにより，プログラミング言語を用いた場合と，構造化チャートを用いた場合の学習効果を比較しようと考えた．

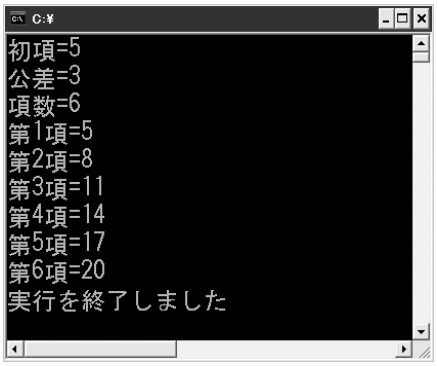


図 7 サンプルの実行結果
Fig. 7 Implementation result of the sample.

なお，事前にアンケート調査を実施した結果，両クラスの学生はプログラミング言語やアルゴリズムを図解化するなどの経験がほとんどなく，両クラスがほぼ等質であることを確認した．

4.1 両クラスの演習の内容

実証実験の対象とした演習科目では，共通してアルゴリズムの学習を目標にしたが，一方のクラスはプログラミング言語（以下，C 言語クラスと記す）を用いて，他方のクラスは構造化チャート（以下，JPADet クラスと記す）を用いて演習を行った．実験の対象とした学生数は，C 言語クラスは 11 名，JPADet クラスは 13 名の合計 24 名であった．**表 1** は期間中に実施した両クラスの学習内容の記録である．各演習時間は 90 分間であり，合計 17 回実施した．なお，C 言語クラスは，StarEdit というエディタ（フリーソフト）でソースプログラムを作成し，LSI-C86 というコンパイラ（フリーソフト）でコンパイルし，コマンドライン上からプログラムを指定して実行するという環境を用いた．

4.2 課題の出題

毎回の演習では，アルゴリズムに関する学習内容の説明と例の提示などを実施した．いくつかの演習（C 言語クラスと JPADet クラスで異なる回数）においては，練習問題を課題として出題した[13]．学生は提示された例の実行確認，および課題の作成と提出を行った．説明に使用した例と練習問題の内容は以下のとおりであり，〈 〉内は課題番号を示す．

(1) 出力，連接処理と数値変数〈1〜5〉
(2) 入力処理と文字変数〈6〜9〉
(3) 選択構造〈10〜12〉
(4) 前判定繰返しと後判定繰返し構造〈13〜15〉

表 1 学習内容の記録
Table 1 Record of the contents of study.

回数	C言語クラス	JPADetクラス
第1回	オリエンテーション，事前アンケート調査，パソコンの使い方	
第2回	C言語とは，ソースファイルの編集方法，コンパイル・実行の方法	流れ図・構造化チャート（PAD）の解説，JPADetの使い方
第3回	出力処理	出力処理
第4回	数値の変数，文字の変数	数値の変数，文字の変数
第5回	入力処理	入力処理
第6回	選択構造	選択構造
第7回	前判定繰り返し構造	前判定繰り返し構造
第8回	後判定繰り返し構造	後判定繰り返し構造
第9回	条件反復構造	条件反復構造，配列
第10回	配列，簡単なアルゴリズム1	簡単なアルゴリズム，選択ソート1
第11回	簡単なアルゴリズム2	選択ソート2，バブルソート1
第12回	選択ソート	夏休み前までの復習，バブルソート2
第13回	バブルソート1	順次探索
第14回	バブルソート2	(出張につき休講)
第15回	順次探索1	バイナリ探索
第16回	順次探索2	これまでの総復習
第17回	最終テスト	

(5) 条件値判定繰返し構造と配列〈16〜18〉
(6) 整列アルゴリズム〈19〜22〉
(7) 探索アルゴリズム〈23〜24〉

なお，課題の提出にあたっては，C 言語クラスでは C 言語のソースとその実行結果，JPADet クラスでは PAD の図とその実行結果を印刷したものの提出を義務づけた．

4.3 アルゴリズムの最終テスト

第 17 回の演習では，両クラスとも最終テストを実施した．なお，テストの条件は，授業中に配布したプリントと資料の持ち込みは可，コンピュータの使用は不可にした．最終テストは，C 言語クラスでは C 言語を用いたアルゴリズムの問題，JPADet クラスでは PAD を用いたアルゴリズムの問題を出題した．一方は C 言語で他方は PAD というように，アルゴリズムの表現方法は異なるものの，問題の質や量は同じになるように設定した．

具体的には，**付録 2** と**付録 3** のような，穴埋め問題（問題番号 1）と作成問題（問題番号 2）の大きく分けて 2 種類を出題した．問題番号 1 が 8 つの設問，問題番号 2 が 6 つの評価ポイント（以下 6 つの設問とする），合計 14 題とした．この評価ポイントとは，以下の 6 項目であり，それぞれの評価基準は次のとおりである．

- 無限ループの有無：C 言語クラスでは while 文か DoWhile 文を使っているかどうか．PADet クラスでは図 4 の②か③を使っているかどうか．
- 分岐構造の扱い：C 言語クラスでは if 文か swich 文を使っているかどうか．PADet クラスでは図 4 の⑤を使っているかどうか．
- カウンタの役割：C 言語クラスと PADet クラスともに，カウンタの役割を担う変数を宣言して，かつ適切に使用しているかどうか．
- 合計計算の扱い：C 言語クラスと PADet クラスともに，合計の計算ができるように変数と数式を記述し，かつそれらを適切な場所に配置しているかどうか．
- 平均計算の扱い：C 言語クラスと PADet クラスともに，平均の計算ができるように変数と数式を記述し，かつそれらを適切な場所に配置しているかどうか．
- 仕様との適合性：C 言語クラスと PADet クラスともに，問題の仕様に適合しているかどうか．

なお，問題番号 1 の穴埋め問題は 1997 年度大学入試センター試験「数学 II・数学 B」の第 6 問で出題された BASIC を用いた問題を，それぞれ C 言語と PAD を用いたものに改題したものである．元々の問題は，BASIC プログラムからしかアルゴリズムの意味が理解できないようにしてあり，解答するためには，アルゴリズムをトレースできることと，整数の除法を理解し正解を絞っていくことの 2 点が必要だと指摘されている[14]．

5. 考　　察

ここでは，4 章で述べた実証実験の考察について述べる．演習の実施スピードや課題の出題状況，最終テストの結果について，プログラミング言語を用いた場合と，構造化チャートを用いた場合を比較して述べる．

5.1 演習の実施スピード

表 1 の期間中に実施した両クラスの学習内容の記録を見ても分かるように，両クラスの演習の実施スピードに差が生じた．具体的には，第 9 回以降からは，JPADet クラスの方が C 言語クラスよりも早く授業内容が進んでいる．その結果，JPADet クラスはバイナリ探索まで進んだのに対して，C 言語クラスは順次探索までで終わっている．また，第 16 回では，JPADet クラスの方は復習を行う時間がとれたのに対して，C 言語クラスではまだ順次探索の演習を行っていた．

以上のような進捗状況の差となる原因の 1 つとして，C 言語と JPADet の仕様の差が考えられる．たとえば，データ型の宣言時には，JPADet では数値と文字の区分しかない．これは VBA（Visual Basic for Application）などのバリアント型ように，データ型を詳細に指定しなくても，変数を利用することが可能な方が，初心者には理解しやすいと考えたためである．VBA のバリアント型までも指定がないのは良くないと考えたため，JPADet の仕様では数値と文字の違いだけを採用した．一方，C 言語では，数値には，[unsigned] int 型，[unsigned] short 型，[unsigned] long 型，float 型，double 型，long double 型があるとともに，それぞれ扱える値の範囲や演算操作も異なってくる．文字には，1 文字と文字列の明確な区別がある．それにともない，入出力関数の指定では，対応する書式指定をそれぞれ指定（%d，%ld，%f，%lf，%d，%u，%c）しなければならない．

しかし，JPADet クラスでは，一般的なプログラミング言語では JPADet のような数値と文字の区分だけではなく，数値では整数と実数の違い，文字では文字と文字列の違いなどを意識する必要があることを解説していた．一方，C 言語クラスでは実際にソースコードの作成において，数値を実数型に，文字を文字列型に限定していた．このように両クラスの進捗状況に仕

様の違いが影響しないような工夫を行っていたので，表 1 を見ても分かるように，第 8 回目までは同じ進捗状況である．

違いが出てきたのは第 9 回目以降であり，進捗状況の差の原因として，配列に関する仕様の違いが考えられる．配列の宣言時に関して，JPADet では何次元でも単純な線形的な並びとして扱えるのに対して，C 言語では各次元を明示的に与えて使用する必要がある．これら C 言語特有の取り決めを，理解し覚える手間がかかることで演習時間がかかる要因になるといえる．しかし，両クラスで実施した演習内容では，1 次元配列のみを扱っており，これらの違いが影響したとは考えにくい．

もう 1 つの原因としては，C 言語が構文中心の文的表象であり，1 次元的にアルゴリズムを表現する必要があることに対して，JPADet は 2 次元的に図を用いてアルゴリズムを表現できることが考えられる．アルゴリズムを考える際に，JPADet は図式であるがゆえに，C 言語よりも直感的に展開することができる[15]．具体的には，PAD の個々のシンボルがプログラムの制御の流れを視覚的に直感できる図式構成になっているのに対し，C 言語ではステートメントが線形的に並んでいるだけの文構成であることから，初心者にとってはアルゴリズムをどのように展開すればいいのかが分かりにくくなる原因になる．PAD では，基本線（main 箱に連結している縦の線）に対して，縦方向にアルゴリズムの全体的な構成を記述したうえで，横方向にアルゴリズム（選択，繰返し）を展開していくことでプログラムを作成することができる．C 言語でも構造化コーディング，すなわちインデントを適切に用いることでアルゴリズムの横方向への展開が可能であるが，そのような工夫を初心者は意識することを忘れがちである．よって，アルゴリズムのデザインの容易さという点からは，構造化チャートの PAD を採用したことが影響し，C 言語に比べて JPADet の方が優位であったと筆者らは考えている．しかし，このことを科学的に裏付けるためには，教育実践をフィールドとした実証実験よりも，実験室における認知心理学的な実験を行う方が望ましく，今後の研究課題とする．

ちなみに，TA（Teaching Assistant）として両クラスに共通して参加していた学生にインタビューしてみたところ，C 言語と JPADet のもう 1 つの仕様の差，つまりプログラム処理系の違いがあるかもしれないという意見があった．JPADet はインタプリタであり，C 言語はコンパイラである．JPADet はインタプリタであることから，翻訳実行は 1 つのシステム上で一度に行えて，初心者にとってプログラム開発環境をほとんど意識せずにプログラムの動作確認ができる．これに対して C 言語は，エディタやコンパイラの操作および実行の仕方などについて意識しなければならないし，毎回一連の操作を繰り返さなければならない．これらの作業に時間がかかることで，C 言語の方はプログラム作成が遅くなることが多かったらしい．ただし，今後については C 言語の統合開発環境にもより簡便な操作性が提供されると思われ，上述のような仕様の差について考慮する必要がなくなることも考えられる．

5.2 課題の出題状況

演習の実施スピードに差が出たため，課題の合計出題数でも異なる結果になった．C 言語クラスでは，合計 12 回の演習において合計 22 題の出題，JPADet クラスでは合計 11 回の演習において合計 24 題を出題した．両クラスの課題の出題状況を比較するために，演習ごとの出題率を求めて累積のグラフに表した．ただし，課題の出題数が影響しないように，それぞれのクラスの合計課題出題数を母数とした．**図 8** に両クラスの課題の出題状況（累積）を示す．JPADet クラスでは，C 言語クラスに比べて，累積出題率が早く 100%に近づいている．しかし，注目するほど際だって大きな差ではなかった．

5.3 最終テストの結果

最終テストの結果を，**表 2** に示す．問題 1 に関しては，両クラスともに低い正解数であった．問題自体は記載されているアルゴリズム意外の既存の数学的な知識からは解答できないように工夫されていることから，解答するときにアルゴリズム自体の理解に時間がかかり，混乱していたことが考えられる．また，両クラスの正解数に差が生じなかったことは，アルゴリズムをトレースできることと，整数の除法を理解し正解を絞っていくことについては，C 言語で学習しても JPADet で学習しても，あまり差がないということが考えられる．

一方，問題 2 に関しても両クラスともに低い正解数であった．作成すべきアルゴリズム自体をイメージできず，混乱していたことが考えられる．しかし，平均正解数が C 言語クラスよりも JPADet クラスの方が明らかに高いことが分かる．アルゴリズムを作成する能力に関しては，JPADet クラスの方が C 言語クラスよりもより身についていることが考えられる．ここでもアルゴリズムのデザインの容易さという点から，1 次元的にアルゴリズムを表現する必要がある C 言語に比べて，2 次元的に図を用いてアルゴリズムを表現できる JPADet の方が優位であると筆者らは考えて

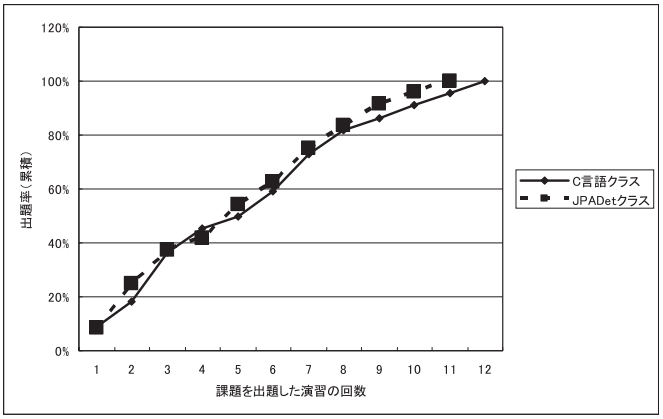


図 8 両クラスの課題の出題状況（累積）
Fig. 8 Comparison of the exercise speed of both classes.

表 2 最終テストの結果（平均正解数）
Table 2 Result of the examination.

	C言語クラス	JPADetクラス
問題番号1(8題)	2.36	2.31
問題番号2(6題)	0.73	1.46
合計(14題)	3.09	3.77

いる．しかし，これらのことを科学的に裏付けるためには，5.1 節と同様に，教育実践をフィールドとした実証実験よりも，実験室における認知心理学的な実験を行う方が望ましく，今後の研究課題とする．

6. むすび

本論文では，筆者達が開発したアルゴリズム学習支援システム JPADet の概要とその実証実験について述べてきた．

今回の実証実験では，C 言語を用いた学習クラスと比較して，JPADet を用いた学習クラスの方がアルゴリズムの演習速度が早いこと，アルゴリズムをトレースする能力には違いが見られなかったが，新規にアルゴリズムを作成する能力は JPADet を用いた学習クラスの方が高いこと，といった結果が得られた．これより，一般情報処理教育における「プログラミング」教育の実施には，プログラミング言語だけの利用よりも，JPADet のようなアルゴリズム学習支援システムの利用を筆者らは提案する．

今後の課題としては，1 次元的にアルゴリズムを表現する必要がある C 言語に比べて，2 次元的に図を用いてアルゴリズムを表現できる JPADet の方が，アルゴリズムのデザインの容易さという点で優位であるかどうかを，科学的に裏付けるために，実験室における認知心理学的な実験を行う必要がある．また，JPADet のシステム自体をさらなる改良（学習履歴情報の収集と解析，採点の自動化，といった機能追加）を目指したいと考えている．それとともに，高等学校の教科情報の「情報 B」における高校生を対象とした利用についても実証実験を進めていきたい．

謝辞 本研究のシステム “JPADet” は，(株) 日本科学技術研修所の営業本部の高田克美氏，佐々木潔氏，香川智彦氏，ならびに先端ソフト開発部の酒井融二氏，佐藤敬之氏のご協力を得たので深く感謝する．なお，本研究の一部は，東京国際大学平成 15 年度特別研究助成（共同研究）の支援を受けた．

参考文献

1) 情報処理学会：大学等における一般情報処理教育の在り方に関する調査研究，情報処理学会 (1993).
2) 文部省（編）：高等学校学習指導要領解説情報編，文部省 (2000).
3) 河村一樹：構造化プログラミングエディタ PADET/CBL の開発と教育での適用，教育システム情報学会誌，Vol.14, No.4, pp.151–160 (1997).
4) Shneiderman, B., Mayer, R., McKey, D. and Heller, P.: Experimental investigations of the university of detailed flowcharts in programming, *Comm. ACM*, Vol.20, No.6, pp.373–381 (1977).
5) 日本規格協会（編）：JIS ハンドブック情報処理ソフトウェア編，日本規格協会 (1997).
6) Warnier, J.D., Flanagan, B.M.: ENTRAINEMENT A LA PROGRAMMATION Tome1-Construction des Programmers, Les Editions d’Organisation, Paris (1971).
7) 二村良彦：プログラム技法—PAD による構造化プログラミング，オーム社 (1984).
8) 矢野将之，藤崎邦博，平嶋 宗，竹内 章：再帰構造の図式を導入した Prolog プログラミング学習支援システム，教育システム情報学会誌，Vol.18,

No.3・4, pp.319–327 (2001).
9) 中村 孝，大垣 斉，高根慎也：図的言語 programa を用いた初級プログラミング教育の試み，平成 14 年度情報処理教育研究集会講演論文集，pp.217–218 (2002).
10) 神村伸一：構造化チャートを利用したアセンブリ言語教育の提案，情報処理学会研究報告，Vol.99, No.17, pp.1–7 (1999).
11) 石田真樹，桑田正行：C プログラミングの学習支援に関する研究，情報処理学会研究報告，Vol.2000, No.117, pp.41–48 (2000).
12) 遠矢行史，川人弘毅，矢澤路朗，鶴巻浩史：職業プログラマー入門，エーアイ出版 (2003).
13) 河村一樹，斐品正照：よくわかる日本語 PAD によるアルゴリズム演習，日刊工業新聞社 (2003).
14) 情報教育学研究会（IEC）編：情報教育からみた入試センター試験（数学），第 9 回情報教育フォーラム配付資料（1997 年から 1999 年の 3 年間にわたって，教育工学関連学協会連合や教育システム情報学会の全国大会で研究発表されたものをまとめた冊子）(1999).
http://www.psn.ne.jp/~iec-ken/
15) 山崎治，三輪和久：図を用いた問題解決，教育システム情報学会誌，Vol.19，No.1，pp.38–45 (2002).

付　　録

A.1 JPADet の文法定義

構文解析にともなう構文規則は，以下のように定義する．ただし，**表 3** のような（BNF に近い）記法を用いる．

処理　process = statement....
命令　statement = assignment|call
代入文　assignment = variable“ = ”expression
変数　variable = ident[“[”expression_list“]”]
式列　expression_list = expression[“,”expression]...
呼出し　call = ident[“(”expression_list“)”]
式　expression = term[opterm]...
2 項演算子
op = “ + ”|“ − ”|“ * ”|“/”|“%”|“^”
|“ < ”|“ > ”|“ = ”|“ <= ”|“ >= ”|“ <> ”
|“&”|“|”
項　term = variable|call|number|string
|uni_opterm|“(”expression“)”
単項演算子
uni_op = “ + ”|“ − ”|“~”
変数列　variable_list = variable[“,”variable]...
条件値判定繰返し
for = variable“ : ”expression_list

なお，上記の中の 2 項演算子の優先順位は，次のとおりとする．

高い　＊　％　^
↑　＋
＜＞　<= >=　<>　＝
↓　&
低い　|

また，構文上のトップレベルは，箱の種類によって**表 4** のように異なるものとする．

A.2 最終テストの内容（C 言語クラス用）

1. 次のような C 言語によるアルゴリズムの記述がある．以下の各設問の空欄を埋めなさい．

```
#include <stdio.h>
void main(void){
        int A,B,Q,R,sw=0;
        printf("A=");
        scanf("%d",&A);
        printf("B=");
        scanf("%d",&B);
        do{
                Q=A/B;          ◀……※
                R=A-Q*B;
                printf("%d¥n",Q);
                if(R!=0){
                        A=B;
                        B=R;
                }
                else sw=1;
        }while(sw==0);
        printf("%d¥n",B);
```

表 3 記法の意味
Table 3 Meaning of description law.

A=B	生成規則(AとはBである)
名称	終端記号か非終端記号
“.....”	終端記号
X...	Xの1回以上の繰返し
[X]	省略可能なX
[X]...	Xの0回以上の繰返し
a|b	aまたはb

表 4 箱と構文の関係
Table 4 Relation between the box and structure.

箱の種類	構文
処理箱	処理
多岐選択	式
前判定繰返し	
後判定繰返し	
条件値判定繰返し	条件値判定繰返し
入力	変数列
出力	式列

(1) 「A=」に対して「12」,「B=」に対して「4」を入力したとき，ディスプレイに数を新たに（ア）個表示して，アルゴリズムの実行を終了する．表示される最初の数は（イ），次の数は（ウ）である．
(2) 「A=」に対して「548」,「B=」に対して「148」を入力したとき，図中の※印の矢印が指す先のところは（エ）回実行してからアルゴリズムの実行を終了する．ディスプレイに表示される最初の数は（オ），次の数は（カ），3 番目の数は（キ）である．
(3) 「A=」に対してはある数 K,「B=」に対して「100」を入力したところ，ディスプレイに表示された最初の 3 個の数は順に「2」,「4」,「2」であった．このとき，ある数 K は（ク）である．

2. 以下の仕様説明文をよく読み，それを実現するアルゴリズムを作成しなさい．なお，作成するアルゴリズムは，C 言語によるプログラムを記述すること．

☆仕様説明文☆

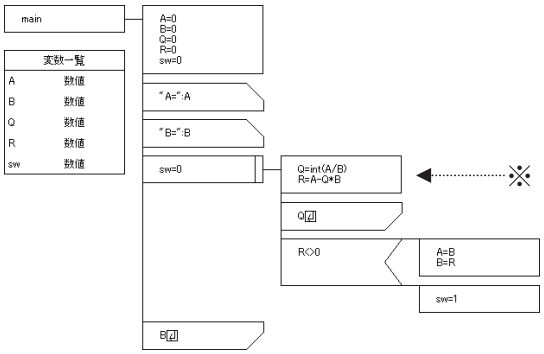
- そのアルゴリズムは，成績（100 点満点）をキーボードから入力するとその評価を表示する．ただし，その評価は，80 点以上は優，70 点以上は良，60 点以上は可，59 点以下は不可とする．
- そのアルゴリズムは,「9999」をキーボードから入力すると実行を終了する．ただし，実行終了時には，それまで入力された成績の合計と平均を表示する．
- 以下にアルゴリズム終了時のディスプレイの状態例を示す．なお，キーボードから入力した数値は「67」,「78」,「82」,「98」,「59」,「9999」である．

成績は？(終了は 9999) 67
可
成績は？(終了は 9999) 78
良
成績は？(終了は 9999) 82
優
成績は？(終了は 9999) 98
優
成績は？(終了は 9999) 59
不可
成績は？(終了は 9999) 9999
合計は 384 点でした．平均は 76 点でした．
実行を終了しました

A.3 最終テストの内容（JPADet クラス用）

1. 次のような PAD によるアルゴリズムの記述がある．以下の各設問の空欄を埋めなさい．

なお「int( )」は小数点以下切り捨てを行う演算である．



(1) 「A=」に対して「12」,「B=」に対して「4」を入力したとき，ディスプレイに数を新たに（ア）個表示して，アルゴリズムの実行を終了する．表示される最初の数は（イ），次の数は（ウ）である．
(2) 「A=」に対して「548」,「B=」に対して「148」を入力したとき，図中の※印の矢印が指す先のところは（エ）回実行してからアルゴリズムの実行を終了する．ディスプレイに表示される最初の数は（オ），次の数は（カ），3 番目の数は（キ）である．
(3) 「A=」に対してはある数 K,「B=」に対して「100」を入力したところ，ディスプレイに表示された最初の 3 個の数は順に「2」,「4」,「2」であった．このとき，ある数 K は（ク）である．

2. 以下の仕様説明文をよく読み，それを実現するアルゴリズムを作成しなさい．なお，作成するアルゴリズムは，PAD による図解表現を用いること．

☆仕様説明文☆

- そのアルゴリズムは，成績（100 点満点）をキーボードから入力するとその評価を表示する．ただし，その評価は，80 点以上は優，70 点以上は良，60 点以上は可，59 点以下は不可とする．
- そのアルゴリズムは,「9999」をキーボードから入力すると実行を終了する．ただし，実行終了時には，それまで入力された成績の合計と平均を表示する．
- 以下にアルゴリズム終了時のディスプレイの状態例を示す．なお，キーボードから入力した数値は「67」,「78」,「82」,「98」,「59」,「9999」である．

成績は？(終了は 9999) 67
可
成績は？(終了は 9999) 78
良
成績は？(終了は 9999) 82
優
成績は？(終了は 9999) 98

優
成績は？（終了は 9999）59
不可
成績は？（終了は 9999）9999
合計は 384 点でした．平均は 76.8 点でした．
実行を終了しました

(平成 15 年 12 月 8 日受付)
(平成 16 年 9 月 3 日採録)

斐品　正照（正会員）

1972 年生．1996 年大阪電気通信大学卒業．1998 年同大学大学院情報工学専攻修了．修士（工学）．宮城大学を経て，現在，東京国際大学商学部情報システム学科専任講師．教育情報工学と情報教育に関する実践と研究に従事．著書『情報科教育法』（彰国社),『よくわかる日本語 PAD によるアルゴリズム演習』（日刊工業新聞社）等．教育システム情報学会（情報教育委員会委員），教育工学会，大学教育学会，認知科学会，教育情報学会，ゲーム学会（評議員），AACE，各会員．

徳岡　健一

1965 年生．1987 年日本大学生産工学部卒業．同年（株）日本科学技術研修所入社．CASE ツール・プログラミング言語処理系関連の開発に従事．

河村　一樹（正会員）

1955 年生．1978 年立教大学理学部卒業，1989 年日本大学大学院理工学研究科博士前期課程電子工学専攻修了．博士（工学）．尚美学園短期大学，宮城大学を経て，現在，東京国際大学商学部情報システム学科教授．情報処理教育，教育工学の研究に従事．情報処理学会（情報処理教育委員会委員，一般情報処理教育委員会幹事，コンピュータと教育研究会運営委員），電子情報通信学会，教育システム情報学会，日本教育工学会各会員．著書に『情報科教育法』（彰国社),『マルチメディア社会と情報教育』（柴峰図書出版）等多数．