

パソコン実習支援のためのデータベースの 設計と活用に関する一考察

加藤 武信・水野 政友

はじめに

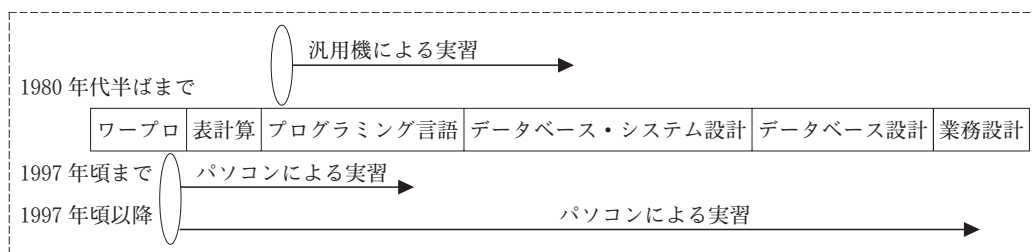
我々は、開発言語 VB (Visual Basic) と関係データベース (Access VBA) を連結したデータベースプログラミングによるパソコン実習支援システムを開発した。それは学生の提出物 (出席票と課題ファイル) を電子的に受け取り、集約し、その提出状況を即座に学生に提示するシステムである。パソコン実習授業でそれを活用した結果、学生の勉学意欲が高まり、実習課題の作成量が従来を上回り、授業の効率化を図ることができた。本稿では、この支援システムを適用した実績分析を紹介する。さらにそのシステムを構成する処理プロセスの概要を説明し、その中の主要な処理については、データベースプログラミングのスキルを盛り込んだプログラムコードを提示する。

1. コンピュータ実習教育の動向とデータベース教育の重要性

1.1 文系学部におけるコンピュータ実習課題範囲の変化

IT (情報技術) の進歩は産業構造を変革しながら情報化社会を進展させているが、そのような変革は大学のコンピュータ実習環境にも恩恵を与え、3~5年毎に繰り返されるパソコンやネットワークのグレードアップの都度ハードとソフトの質と量を著しく拡充させてきた。そして大幅な教育割引制度をもつ高価なハイテクソフトの導入に加えて種々のソフトの無料トライアル版やオープンソースソフトを容易に入手できる機会が増大していることなどが、パソコン利用環境構築の自由度を高めている。

第1図は、これまで多くの文系学部で実施されてきたコンピュータ実習課題の対象領域とその到達レベル (矢印が右に伸びるほど課題レベルが高い) をラフに表したものである。すなわち1980年代半ばまでの情報関連科目では、主に汎用機環境でプログラミング言語からシステム設



第1図 コンピュータ実習課題領域の変化

計に及ぶ課題に取り組む実習が主流であった。しかし80年代後半より高性能かつ安価な16ビットパソコンが汎用機に代替する傾向が進み、パソコンの豊富な応用ソフトを活用してインターネットでのWebページ検索、ワープロ、プレゼンテーションおよび表計算等のソフトを電子文具レベルで活用することが主流になり、プログラミングやシステム設計の学習は下火になっていった。そしてインターネット時代の到来とともに90年代半ばにマイクロソフト社がリリースしたWord, Excel, AccessをセットにしたOffice 95の爆発的普及により電子文具レベルでのパソコン利用の大衆化が一段と進んだ。その後Officeはバージョンを順次97, 2000, 2002 (XP), 2003に上げながら今日に至っているが、Office 97のExcelとAccessにVB (Microsoft Visual Basic Programming System)と同族のVBA (Visual Basic for Application Edision; 応用ソフト版VB)を標準装備 (Office 2000ではWordにも装備)したために、それらは個別業務をシステム化するための本格的な開発環境をもつようになり、電子文具のレベルを超えたツールに進展した。そして今日、WS (ワークステーション)で利用されてきた各種のハイテクソフトのパソコンへの移植も進展するにつれて、データベース設計や業務設計等もパソコン実習領域に入ってきている。

大学教育でのパソコン実習課題は、初心者用から高度なスキルまでの課題を間断なく含む連続した体系をもつことが理想である。そして今日、データベースソフトが多くのパソコンにインストールされエンドユーザのツールとなっている現状から見ても、入門レベルの課題を素早く体験した後に第1図に示すデータベース⁽¹⁾以降の領域を扱う課題が大学における情報処理教育の中心になってよいと考える。

1.2 実習課題の体系化

第1表は、最近6年間におけるパソコン実習課題の設定状況を示す。本学にWindowsパソコンが導入された1996年度以来、その第1ステップとしてインターネット検索、電子メール、WordおよびExcel等を利用した課題を設定したのは当然であるが、多くの受講生がそのレベルに留まり、今後のソフト開発力の養成に熱意を示す学生がきわめて少数であったことに不安を

第1表 最近6年間のパソコン実習課題の設定状況

年度	(○：受講生全般を対象 △：ゼミなど一部の学生を対象)							プログラミング言語					
	Word	Excel	Access	システム設計	DB 設計	業務設計	会計ソフト	VBA	VB	C	C++	J++	Java
1999	○	○	△	△			△	○					
2000	○	○	○	△	△		△	○					
2001	○	○	○	△	○		△	○					
2002	○	○	○	△	○		○	○		△	△		
2003	○	○	○	△	○		○	○	○	○	○	○	
2004	○	○	○	△	○	△	△	○	○	○	○	○	○

抱き、次へのステップアップを図るための対策の必要性を痛感した。

第2ステップであるプログラミング言語を取り扱ったのは、Excel 97 による VBA であった。長年にわたり SE 志望の学生に対して COBOL プログラミングを指導してきた者にとって、本格的な開発環境をもつ Excel VBA はソフト開発力を養成できる有効なツールであると判断したので、早速その教材化を試み 1998 年度から実習を開始した。そして VBA 実習を経験した受講生の興味と達成感の強さは予想を上回ったので、2 年掛かりで一連の課題体系を完成させた。それに加えて他の言語も 2003 年度より VBA と同一のアルゴリズムによるプログラムを作成する方法で実習を開始した。今後は、各言語ともデータベースを操作するデータベースプログラミングのレベルの課題を扱うことを目指している。

第3ステップである Access・システム設計のレベルの課題に関しては、1998 年度より Access 97 を用いて実習を開始した。アプリケーション開発におけるシステム設計は、業務内容の分析とシステム化範囲の特定、入出力設計、データベース設計、画面設計・出力設計の順序で進められるが、その中心はデータベース設計であり、各レベルの設計内容がデータベース上に反映される。そして Access には、基本機能、マクロ、SQL および VBA の 4 つの設計機能があり、それらは単独あるいは 2 つ以上を連係して活用することができる。そしてその基本機能だけでも本格的な業務用アプリケーションを容易に構築できるが、Access を実際に稼働させている SQL のレベルまで掘り下げて学習すれば、データベース構成の原理を理解することができる。さらに VBA を使ったプログラミングを付加すれば、高度な利用技術を習得しアプリケーションの完成度も著しく高めることができる。

第4ステップのデータベース (DB) 設計・業務設計⁽²⁾ レベルにおいて、前者ではコンピュータ・アソシエイツ社の ERwin を用いて 2000 年度よりデータベース設計を開始し、課題内容も年々充実させている。同社の BPwin を用いる後者については、現時点ではその本格的な実習態勢は確立していないものの、今後拡充すべき分野であると期待している。

2. 実習教育支援システムの概要と適用効果

2.1 実習教育支援システム設計までの経緯

パソコン実習授業において発生する受講生の提出物（出席票と提出課題ファイル）をデータベース上に集計し、受講生の要求に応じて即座に提示するシステムについて考察を行う。それは、①受講生の提出物をネットワークを介して教員フォルダに送信する、②教員フォルダに集積された提出物のファイル名をデータベースに登録する、③受講生の要求に応じて該当する提出状況をネットワークを介して提示し、④実習実績に関する全体あるいは部分情報を担当教員に提供するという4つの機能をもつ。

パソコン実習授業ごとに発生する出席票の処理は、受講生が多数であるほど膨大になる。また課題の提出においても、課題数が多くなるほどその管理の負担は重くなる。その解決策として1998年度にパソコン環境での出席と課題提出の電子化（手操作）を実施したが、10%程度の提出ミスが発生した。その対策として2003年度より支援ソフトを開発して出席と課題提出の自動化に取り組み、ミスの防止と処理の効率化を実現した。

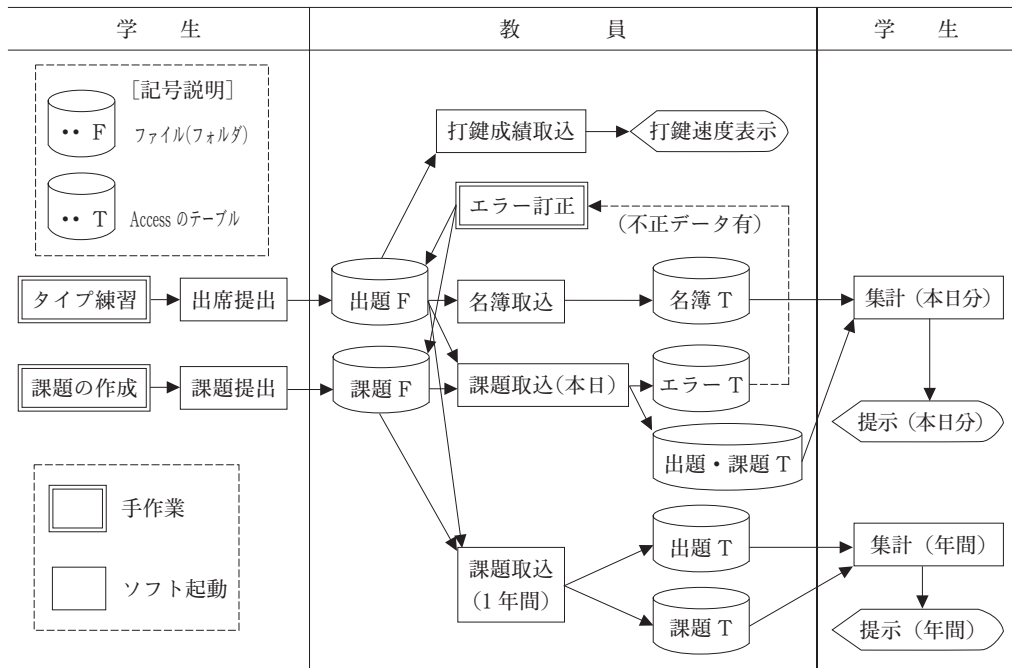
この支援ソフトは、情報関係の講座担当者である加藤（以下Aという）と水野（以下Bという）が開発した。BはAの学部ゼミに所属した時期にTA（Teaching Assistant）としてコンピュータ実習授業に参画した経験がある⁽³⁾。Aは、1981年度より継続して1コマ200人前後の受講生を擁するコンピュータ実習授業に携わる過程で、効果的授業法を追求してきた。そしてその年度から17年にわたり実施した汎用機によるプログラミング実習においては、個別学習方式による実習態勢の構築において、その基本要件である課題の体系化、提出方法の電子化、TAを含めた知識獲得の場の設定を充実させることが学生の勉学意欲を高めることを確認したが、その詳細については複数の論文にまとめている⁽⁴⁾。またAは、1年次～3年次配当の複数の情報科目を担当したので、複数科目に共通する課題を設定し、またゼミ生をTAとして依頼することが容易に行えた。そして、1995年度以降から今日までのパソコン環境での実習においても、これまでの経験を踏まえて次の5つの基本要件をベースにした実習態勢を構築してきた。それに関する考察の成果は既に公表しているが、ここではその一部を掲載する⁽⁵⁾。

- ① 課題体系ベースの解題と課題管理表による自主管理
- ② 技術的トラブル解決のためのTAの役割
- ③ 電子出席票と電子課題の提出とその提示
- ④ タイピング訓練の日常化と訓練実績の出席への組み入れ
- ⑤ 支援ソフトによる出席票と課題の提出とその提示

2.2 実習支援システムの構成

第2図は、実習支援システムの処理概要を示すシステム流れ図であるが、それは①学生が出席票と課題を提出する部分、②教員がそれを処理しデータベースに集約する部分、および③提出状況を学生が閲覧する部分を表示している。

第2表の7つの処理プロセスには、それぞれ開発言語の種類が示されている。その表中の「コード」欄に「表示」という文字があるのは、後に処理内容をプログラムコードで表示することを意



第2図 実習支援システムのシステム流れ図

第2表 実習支援システムの主要な処理プロセス

処理名	コード	開発言語の種類	摘要
①出席票提出処理		Visual Basic	学生フォルダから共通フォルダに出席票を提出
②課題提出処理	表示	Visual Basic	学生フォルダから共通フォルダに課題ファイルを提出
③受講者名簿更新処理	表示	Access VBA	出席票の学籍番号と講義名から受講者名簿を更新
④本日分提出物集計処理		Access VBA	本日分の提出物を集計
⑤年間提出物集計処理	表示	Access VBA	年間の提出物を集計
⑥本日分提出物提示		Visual Basic	個々の学生の要求に応じて本日分の提出物を提示
⑦年間提出物提示処理	表示	Visual Basic	個々の学生の要求に応じて年間の提出物を提示



第3図 実習支援システムの Access テーブル構成と「エラー説明」テーブルの内容

味する。ただし第2図の「打鍵成績取込」処理は現在稼動状況にあるものの、今回の説明から除外しているので第2表には掲載していない。

第3図の画面aは、実習支援システムで用いるデータベース Access の6つのテーブル構成を示している。また画面bの「エラー説明」テーブルの内容は、提出物の正当性チェックで不正データと見なされるエラー原因とその説明を示す。

2.3 拡充ポイントを反映した課題提出実績の分析

第3表は、最近4年間の拡充ポイントを年度ごとにまとめたものである。パソコン実習授業では、パソコンのハード上の不具合、操作ミスあるいは受講者の理解不足等により様々なトラブルが頻発したり質問が多発するが、実習支援システムはそれを解決するための有効な対応策として2001年度より開発に着手した。それは、電子出席と電子課題の提出操作を自動化し、さらに提出物の状況を学生各自のパソコン画面で何時でも閲覧できる機能を備えたが、その開発が完了したのは2003年度である。そのソフトの活用により勉学意欲の高揚を招き提出数が大幅に向上したが、それを招いた理由は、授業時の提出物をその時間中に5回ほど更新したために受講者が最大20分のタイムラグで個々の提出状況を確認できるようになり、解題の達成感を感じ取るようになったことが主因であると考えられる。

実習支援ソフトの活用によって提出物の標準化が徹底し、その最新の提出状況を提示する機能

第3表 実習システムの拡充ポイント

年 度	提出状況閲覧	キーボード訓練	電子出席の提出	電子課題	課題管理表
2001年度前期	授業中閲覧	5～15分程度	手操作	手操作	記入のみ
“ 後期	一覧表示	出席と一体化	“	“	提出義務
2002年度	“	“	“	“	“
2003年度	ソフト利用	“	ソフト利用	ソフト利用	“
2004年度	“	“	“	“	“

により受講者の授業に対する信頼性は深まり、課題管理表をベースにした個別学習方式の完成度は一段と高まった。そのシステムを適用した効果について、授業科目で検証しよう。

2001年度より1年次リテラシー講座である「情報技術入門（仮称）」が半期科目で開設され、Aが担当した。第4表は、年度別実習者数（出席票とExcel課題の双方を1回以上提出した1年次のみ受講者）である。その講座は、3年間ほど必修科目であったが、4年目から選択科目に変更になった。

この授業の目標は、パソコン実習経験の有無に係わらずWordとExcelを中心とする基礎スキルを習得し、その後の上位スキル習得の前提条件を整備することである。

上掲の各年度ごとの拡充ポイントは、実習システムを構成する基本要件を有機的に強化し、実習生の課題提出意欲を増大させたと考えるが、その最大の拡充は実習支援ソフトの開発であった。それによるスキルアップ効果を、この授業の中核であるExcelの提出状況を分析することによって評価してみたい。

第5表は、実習者の課題達成度合をExcelの課題提出数によって4つのランクに分けたものである。ランク分けには実習状況が最も良好であった2003年度をモデルにし、おおよそS、A、B、Cランクの人数がそれぞれ10%、10%、50%、30%になるようにした。

第6表は、各年度ごとのExcel課題の提出状況を実習者のランク別に表したものである。また第3図は、Excel提出課題提出者の課題数別累積割合を示す。

第6表と第4図から、次のことを確認できる。

第4表 情報技術入門の年度別実習者数

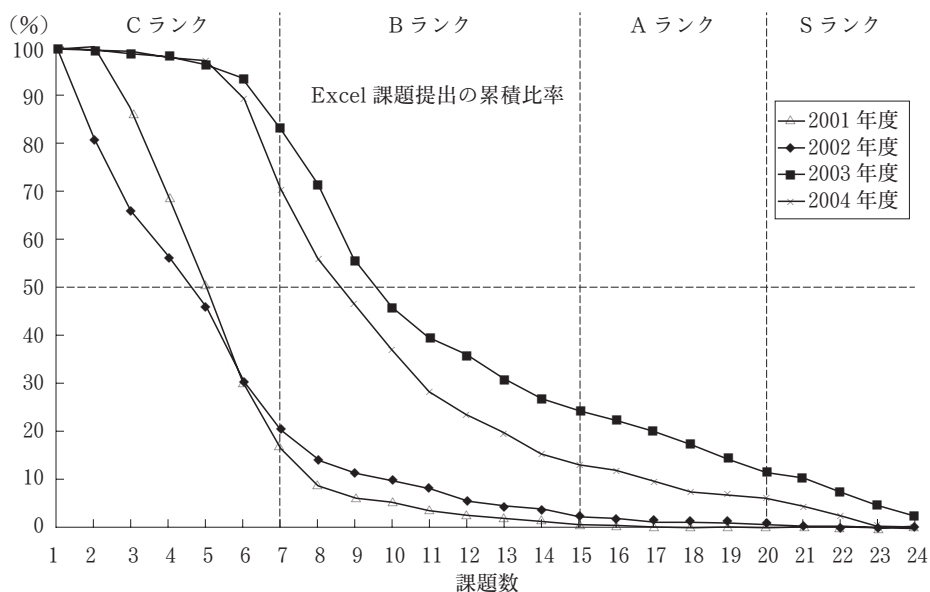
年度	実習者数	コマ数	必修／選択
2001	568人	6	必修
2002	610人	5	〃
2003	591人	5	〃
2004	252人	2	選択

第5表 Excel課題数による4つのランク

種 別	課題範囲
Sランク（自己展開型）	20～24個
Aランク（高目標型）	15～19個
Bランク（中目標型）	7～14個
Cランク（低目標型）	1～6個

第6表 ランク別Excel課題提出状況

年度	実習者数	Sランク	Aランク	Bランク	Cランク	合計
2001	568人	0.0%	0.4%	6.2%	93.5%	100%
2002	610人	0.2%	1.5%	12.3%	86.1%	100%
2003	591人	10.2%	12.0%	48.7%	29.1%	100%
2004	252人	4.4%	7.5%	44.0%	44.0%	100%



第4図 Excel 課題提出状況

- ① 2001年度と2002年度は支援ソフトの開発以前であるが、同じ傾向の曲線である。
- ② 2003年度と2004年度は支援ソフトを使用した後であるが、同じ傾向の曲線である。

支援ソフトの開発以前とそれ以後の状況を、授業条件が最も類似している2002年度と2003年度で比較すると、次のことが指摘できる。

③ 支援ソフトの使用により出席と課題提出状況の提示を徹底したことが、実習生の課題作成意欲を高め、Cランクを55%も減少させ、上位ランクに移行させている。すなわちB, A, Sの各ランクをそれぞれ36.4%, 10.5%, 10%増大させたことは、大きな成果である。

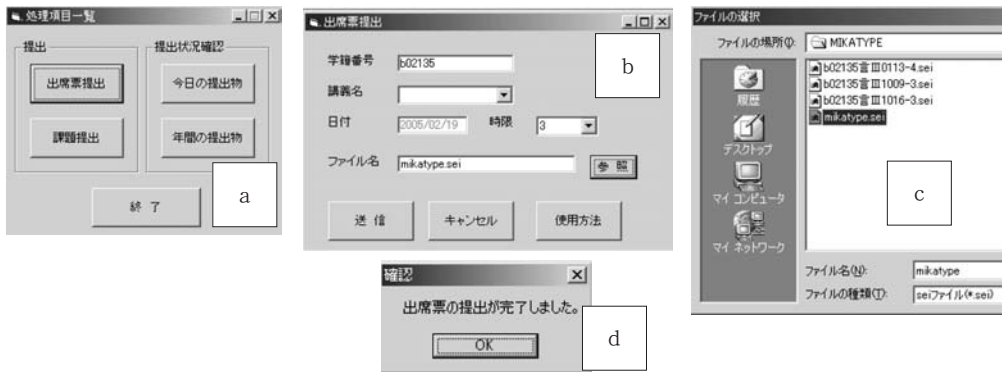
さらに2003年度に比較して2004年度の課題提出数の減少が見られるが、それは、授業時間外の課題提出が2003年度より少なかったためであり、授業時間内の課題作成数はほぼ同じであったように感じている。

3. システムの主要処理の概要

第2表に示す7つの処理のそれぞれについて、その概要を処理画面と関連づけて説明しよう。処理概要そのものは、箇条書きの処理事項を構造化方式で表現し、処理全体を点線で囲んである。そして大括弧で囲んだ処理名は、イベント（ボタンのクリックなどの行為）駆動による処理単位を意味する。また上位=下位の関係は、左側を上位とする「字下げ」によって示してある。

3.1 出席票提出処理（開発言語＝VB）

これは、学生が参加する授業の時間ごとに提出する「電子出席票」を受け取る処理である。学生が自分で管理するフォルダに格納した打鍵成績ファイル（mikatype.sei）を画面 c のように指定すれば、支援ソフトがそれを「学籍番号・講義名・期日－時限」で構成される出席票に自動的に変換して教員が管理する「課題提出」フォルダに提出する（第 6 図画面 d 参照）。



第 5 図 出席票提出処理に用いる画面

[出席票提出] 第 5 図画面 a の「出席票提出」をクリック

① 画面 b を表示し、学籍番号、日付および授業時限を設定

② 講義名をコンボボックスに設定

[講義名] コンボボックスにおいて講義名一覧より選択

[授業時限] 授業時限を変更する時は、コンボボックスで変更可能

[ファイル名] 出席票になる打鍵成績ファイル名とそのフォルダを選択

① [参照] ボタンを押し、フォルダを特定

② 画面 c で打鍵成績ファイルを選択すると、そのファイル名を画面 b に表示

③ 出席票までのパスを設定

[送信] 画面 b の「送信」ボタンをクリック

① 学籍番号、講義名、ファイル名をチェックし、正常でなければメッセージを表示

② 学籍番号・講義名・日付データから出席票のファイル名を標準名に変更

③ 送信元の出席票を標準名に変換

④ 「課題提出」フォルダに出席票をコピー

⑤ 「出席票の提出が完了しました。」を表示

[キャンセル] 処理の中止（画面を b 消去し a 画面を表示）

[使用方法] 出席票提出に関する操作方法を説明

[例外処理] 設定が正常でない場合のエラー処理

3.2 課題提出処理（開発言語=VB）

それは、学生フォルダに格納された課題ファイルを、「学籍番号・標準課題名」の提出課題ファイルに変換した後に教員管理の「課題提出」フォルダに提出する処理である。第6図の画面bの例では、学生が命名した課題名「成績管理修正」を学籍番号・標準課題名である「成績管理」に変換し、画面dの「提出課題」フォルダに提出したことが確認できる。



第6図 課題提出処理で用いる画面

[課題提出] 第6図画面aの「課題提出」をクリック

① b画面を表示し、学籍番号を設定

② 使用ソフト名と課題名をコンボボックスに設定

[使用ソフト名] 提出課題のソフト種類をコンボボックスのソフト名一覧より選択

[課題名] コンボボックスの課題名一覧より提出課題の標準課題名を選択

[ファイル名] 提出課題名とそのフォルダを選択

① 画面bの「参照」ボタンを押して、フォルダを特定

② 提出課題ファイルを選択すると、そのファイル名を画面bに表示

[送信] 画面bの「送信」ボタンをクリック

① 使用ソフト名にソフト名が選択されていることをチェック

② 学生フォルダにある提出課題名を標準課題名に変換

③ 「課題提出」フォルダに提出課題ファイルをコピー（コピー結果を画面dに示す）

④ 「提出が完了しました。ほかのファイルも提出しますか？」を表示（画面c）

[キャンセル] 処理の中止（画面bを消去し画面aを表示）

[使用方法] 課題提出に関する操作方法を説明

[エラー処理] 設定が正常でない場合メッセージを表示

3.3 受講者名簿更新処理（開発言語＝Access VBA）

本処理は、「課題提出」フォルダに蓄積した「出席票」から「受講者名簿」を作成する処理である。第7図の画面aは、「課題提出」フォルダの内容をそのサブフォルダの「提出物全体」フォルダに移動したことを示す。画面bの参照ボタンにより「提出物全体」フォルダの所在を確定した後に「開始」ボタンを押せば、そこに格納された出席票から「学籍番号」と「講義名」を抽出し、画面dの受講者名簿と照合し、未登録であれば追加登録を行う。



第7図 受講者名簿更新処理で用いる画面

〔受講者名簿更新処理〕第7図画面bの「参照」をクリック

「提出物全体」フォルダまでのパスを選定し、その結果を「パス名」欄に表示

〔開始〕正しいパスか否かをチェックし、そのフォルダを検索

提出物全体フォルダを検索し、格納されているファイル名を取り込む

データベースと各テーブルを設定

取り込んだ個々の提出ファイル名を判定し、出席票のみを抽出

学籍番号が受講者名簿に未登録ならば、学籍番号と講義名の双方を登録

学籍番号と講義名の双方が登録されていれば、何もしない

学籍番号は名簿に登録されていて講義名のみ未登録ならば、講義名を登録

学籍番号の登録件数をメッセージとして表示（画面c）

3.4 本日分提出物集計処理（開発言語=Access VBA）

本処理は、「課題提出」フォルダ（第6図画面d）にある本日提出分データをサブフォルダ「提出物本日」に移動した後、そのサブフォルダから出席票と提出課題ファイルを取り出し、データベース内の「提出物仮保存」テーブルに格納する。第8図画面aの[OK]ボタンを押せば、第6図画面dの2個の提出物は第8図画面cのように「提出物仮保存」テーブルに格納されることを示す。

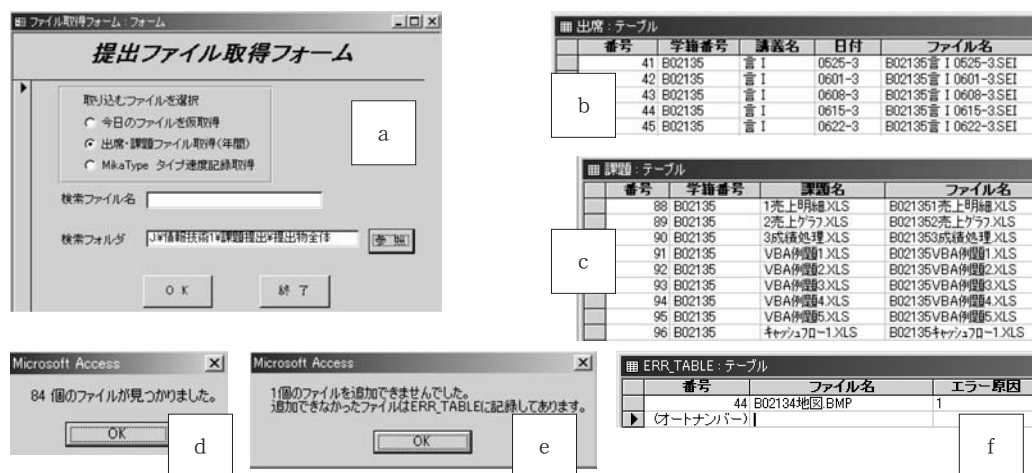


第8図 本日分提出物集計処理で用いる画面

[本日分提出物集計処理] 第8図画面aの「今日のファイルを取取得」を選択し[参照]を押す
「提出物本日」フォルダまでのパスを選定し、そのパスを画面aに表示
[OK] 正しいパスか否かをチェックし、そのフォルダを検索
「提出物本日」フォルダを検索し、格納されているファイル名を取り込む
データベースと各テーブルを設定
取り込んだファイルが正常であればデータベース内の「提出物仮保存」テーブルに格納
個々のファイル名の中の「学籍番号」が受講者名簿に登録済みか否かを判定
名簿登録済みの出席票と課題ファイルは「提出物仮保存」テーブルに格納
名簿未登録の出席票と課題はエラーテーブルに格納しその件数をカウント
エラーテーブルへの格納件数のメッセージを表示

3.5 年間提出物集計処理（開発言語＝Access VBA）

この処理は、年間の提出データである「提出物全体」フォルダに格納した出席票と提出課題ファイルを読み込み、正常なデータであればデータベースの「出席」テーブルと「課題」テーブルに、正常でない場合は「エラーテーブル」に格納する。第7図画面aのデータを集計するために第9図画面aの[OK]ボタンを押せば、画面bとcの「出席」および「課題」テーブルを作成する。なお画面eとfは、不正データの処理を示す。

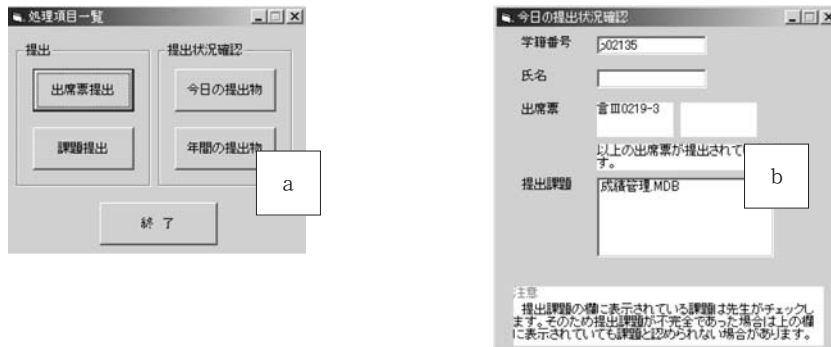


第9図 課題提出処理で用いる画面

[年間の提出物集計処理] 第9図画面aの「出席・課題ファイル取得(年間)」を選択
 [参照]をクリック
 第7図画面a「提出物全体」フォルダまでのパスを選定
 パスを第9図画面aに表示
 [OK] 正しいパスか否かをチェックし、そのフォルダを検索(第9図画面d)
 「提出物全体」フォルダを検索し、格納されているファイル名を取り込む
 データベースと各テーブルを設定
 取り込んだ全部のファイルをデータベース内のテーブルに格納
 個々のファイル名の中の「学籍番号」を判定
 名簿登録済みの出席票の場合は出席テーブルへ格納(第9図画面b)
 名簿登録済みの課題の場合は課題テーブルへ格納(第9図画面c)
 名簿未登録の出席票と課題は、エラーテーブルへ格納(第9図画面f)
 エラーテーブルへ格納した件数を表示(第9図画面e)

3.6 本日分提出物提示処理（開発言語=VB）

それは、データベース内にある「提出物仮保存」テーブルに存在する本日分データを学生の要求に応じてユーザ画面に提示する処理である。第10図画面aで「本日の提出物」ボタンを押せば、即座に画面bが提示される。



第10図 本日分提出物提示処理で用いる画面

[本日分提出物提示処理] 第10図画面aの「今日の提出物」ボタンをクリック
 学籍番号を取得し、データベースの「受講者名簿」テーブルを検索（第7図画面d）
 該当する学籍番号が存在しない場合、メッセージを表示
 データベースの「提出物仮保存」テーブルを検索（第8図画面c）
 該当する学籍番号のデータが存在する場合
 出席区分レコードを抽出し、第10図画面bの「出席票」欄に表示
 課題区分レコードを抽出し、第10図画面bの「提出課題」欄に表示
 該当する学籍番号のデータが存在しない場合、メッセージを表示

3.7 年間提出物提示処理（開発言語=VB）

年度初めから蓄積された提出物は、データベース内の「出席」テーブルと「課題」テーブルに格納されている。第11図の画面aで「年間の提出物」ボタンを押すことにより、出席状況と課題提出に関するデータが集計加工され、第11図画面bに提示される。



第 11 図 年間提出物提示処理で用いる画面

[年間提出物提示処理] 第 11 図画面 a の「年間の提出物」ボタンをクリックすると、順次以下の処理を行う。(ただし長方形で囲った処理は、サブプロシージャまたは関数（ファンクションプロシージャ）であることを示す。)

データベースより学籍番号のレコードセットを取得 (GetPastData)

パソコン使用者の学籍番号を取得

データベースとのコネクションを設定

学生情報を取得 (GetStudentInfo)

「学籍番号 SQL を生成」(値が Null ならば処理をしない)

同一学籍番号のレコードセットを構造体に格納

出席データを取得 (GetAttendData) (第 9 図画面 b)

「講義別出席 SQL を生成」(値が Null ならば処理をしない)

レコードセットを構造体に格納

課題名データを取得 (GetKadaiData) (第 9 図画面 c)

「課題名 SQL を生成」(値が Null ならば処理をしない)

レコードセットを構造体に格納

拡張子別課題数を集計 (GetKadaiCount)

拡張子別課題数を集計

集計結果を構造体に格納

指定した拡張子以外の課題を集計

構造体に格納されたデータをフォームに提示 (SetFormData) (第 11 図画面 b)

学生情報を画面 b に提示

出席データを画面 b に提示

課題名データを画面 b に提示

拡張子別課題数を画面 b に提示

指定した拡張子の課題数の合計を画面 b に提示

4. 主要処理のプログラムコード表示

実習支援ソフトの7つの処理それぞれの概要を上掲のように説明したが、第2表のコード欄に「表示」と記してある課題提出処理、受講者名簿更新処理、年間提出物集計処理、年間提出物提示処理の4つについては、さらに開発言語のプログラムコードを表示して詳細な処理の内容を検討する。

4.1 課題提出処理 (VB)

これは、第6図画面aの「課題提出」画面を表示した後、ユーザの学籍番号を設定し、関数 mdlData (後述) より取得した使用ソフト名と課題名をコンボボックスに格納する処理を行う。

(1) 画面aの「課題提出」ボタンにより開始される処理

```
Private Sub Form_Load()
    Dim varSoftName As Variant, varKadaiName As Variant, v As Variant
    Dim i As Integer
    txtNumber.Text=mdlData.GetNumber           '学籍番号をフォームに設定
    varSoftName=mdlData.GetSoftName           'mdlDataより使用ソフト名を取得
    For Each v In varSoftName
        If v<>"" Then
            cmbSoftName.AddItem(v)           '使用ソフト名をコンボボックスに格納
        Else
            Exit For
        End If
    Next
    varKadaiName=mdlData.GetKadaiName         '標準課題名を mdlDataより取得
    For Each v In varKadaiName
        If v<>""Then
            strKadaiName(i)=v                '標準課題名をコンボボックスに格納
            i=i+1
        Else
            Exit For
        End If
    Next
End Sub
```

(2) 第6図画面bで「使用ソフト名」と「課題名」を選択した後の処理

「使用ソフト名」を選択した後、関数 ComCtl32Bas を用いてその拡張子に限定した標準課題

名をコンボボックスに格納する。

```
Private Sub cmbSoftName_LostFocus()
    Dim strSoftName As String, i As Integer
    If cmbSoftName.Text<>""Then      '提出課題の使用ソフト名の選択を確認
        cmbRefarence.Enabled=True    '参照ボタンを ON
        txtFilePath.Enabled=True     'パス設定を ON
        cmbKadaiName.Clear           '課題名エリアをクリア
        i=0
    Do Until(strKadaiName(i)="")
        '選択されたソフト名に限定した標準課題名をコンボボックスに格納
        If InStr (strKadaiName(i), "=")<>0 Then
            strSoftName=Left(strKadaiName(i), InStr(strKadaiName(i), "=")-1)
            If strSoftName=cmbSoftName.Text Then
                cmbKadaiName.AddItem (Mid(strKadaiName(i),
InStr(strKadaiName(i), "=")+1))
                End If
            End If
            i=i+1
        Loop
    End If
End Sub
```

(3) 第6図画面bの[参照]ボタンを押した後、フォルダを次々に選択してパスを確定する。さらにフォルダの中で選択したソフト名の拡張子を元に ComCtl32Bas (後述) の機能を使って「使用ソフト名」で示す拡張子をもつファイル名のみを表示する。

```
Private Sub cmbRefarence_Click()
    Dim strSoft As String
    Dim objFso As Object
    Set objFso=CreateObject("Scripting.FileSystemObject")
    Select Case Left(cmbSoftName.Text, 4)
        Case "ペイント": strSoft="*.bmp"
        'ユーザのフォルダに格納したファイルの中から特定の拡張子をもつファイル名のみを抽出
        Case "Word": strSoft="*.doc"
        Case "Exce": strSoft="*.xls"
        Case "Powe": strSoft="*.ppt"
        Case "Acce": strSoft="*.mdb"
        Case "その他": strSoft="*.*"
    End Select
```

```

txtFilePath.Text=Select_OpenFileName(Me.hWnd, "ファイルの選択", "I:¥", strSoft)
txtFilePathDmy.Text=objFso.GetFileName(txtFilePath.Text)
Set objFso=Nothing
End Sub

```

(4) 第6図画面bの「送信」ボタンを押して開始される処理

使用ソフト名、標準課題名、提出ファイル名が選択されていることを確認した後、提出課題名を標準名に変換する。さらに FileSystemObject を使用し、そのファイルを「課題提出」フォルダにコピーする。

```

Private Sub cmdTrams_Click()
    Dim objFileSystem As Object, intPathLen As Integer
    Dim strFileName As String, strNewFilePath As String
    Dim intExtesionLen As Integer
    Set objFileSystem=CreateObject("Scripting.FileSystemObject")
    '使用ソフト名、標準課題名、提出ファイル名が選択されていることを確認する判定処理
    intExtesionLen=InStrRev(txtFilePath.Text, ".", , vbTextCompare)
    strFileName=txtNumber.Text & cmbKadaiName.Text & Mid(txtFilePath.Text, intExtesionLen)
    '提出課題名を標準名に変換
    intPathLen=InStrRev(txtFilePath.Text, "¥", , vbTextCompare)
    'FileSystemObject を使用し、setting ファイルの提出先パスへファイルをコピー
    Name txtFilePath.Text As strFileName
    strNewFilePath=Left(txtFilePath.Text, intPathLen)& strFileName
    objFileSystem.CopyFile strNewFilePath, mdlData.getTramsPath, False
    If MsgBox("課題が提出が完了しました。ほかの課題ファイルも提出しますか?", _
        vbYesNo, "提出完了")=vbYes Then
        txtFilePath.Text=""
        GoTo END 1
    Else
        Unload Me
        Exit Sub
    End If
End Sub

```

(5) 関数「mdlData」のプログラムコード

「mdlData」は、Windows 内部から取得する環境情報（学籍番号、日付、時刻等）と設定ファイル「setting.ini」に格納された授業情報とを内部で保持し、他のプログラムに引き渡すモジュールである。「mdlData」が読み込む「setting.ini」ファイルの構成を第12図に示すが、それは教

員が設定する実習形態とパソコン環境に関する規約事項（講義名，使用ソフト名，課題名，提出先パス名等）から構成される。

//出席フォームに関する設定 [担当講義名]	ペイント=富士山 ペイント=地図	Access=販売仕入 Access=販売仕入 VBA
講義名=技術	Word=メニュー表	Access=成績管理
講義名=言Ⅰ	Word=手紙	・
講義名=言Ⅲ	Word=流れ図	言語=Jmon 11
講義名=処理	Word=案内	言語=Jmon 12
・	・	言語=Jmon 13
//課題フォームに関する設定	Excel=課題管理表	言語=Jmon 14
[使用ソフト名]	Excel=円グラフ	言語=Jmon 15
ソフト=ペイント	Excel=複合グラフ	言語=Jmon 21
ソフト=Word	Excel=おでん表	ERwin=成績管理元
ソフト=Excel	Excel=抽出条件	[提出先パス]
ソフト=Power Point	Excel=各種関数	提出先=J:¥情報技術1¥課題提
ソフト=Access	Excel=大声コンテスト	出¥
ソフト=言語	Excel=基本関数	[DBパス]
ソフト=ERwin	・	参照元=J:¥情報技術1¥教材配
・	Power Point=研究会	布¥提出物管理システム.
[課題提出名]	Power Point=練習 1	mdb
ペイント=バンド	Power Point=練習 2	

第 12 図 「setting.ini」 ファイルの構成

```
Private Declare Function getUserNam& Lib "advapi32.dll" Alias "GetUserNameA"(ByVal
NameBuff As String, Size As Long)
Dim strNumber As String, strLecture(6) As String
Dim strSoftName(10) As String, strKadaiName(256) As String
Dim strTramsPath As String, strRefarencePath As String
Sub setData()
    Dim strUserDmy As String*255, strDat As String
    Dim strFileName As String
    Dim intFileNo As Integer, v As Variant
    Dim i As Integer, k As Integer, j As Integer
    getUserNam strUserDmy, 255
    For i=1 To 255 '文字列を VB 用に処理
        If Asc(Mid$(strUserDmy, i, 1))=0 Then
            strNumber=Mid$(strUserDmy, 1, i-1)
            Exit For
        End If
    Next
    i=0
    strFileName="setting.ini"
    intFileNo=FreeFile
```

```

Open strFileName For Input As#intFileNo
Do Until EOF(intFileNo)
  Line Input#intFileNo, strDat
  If blnFlg=True Then
    strKadaiName(i)=strDat
    i=i+1
  End If
  Select Case Left(strDat, 4)
    Case "ソフト=": strSoftName(k)=(Mid(strDat, 5))
      k=k+1
    Case "講義名=": strLecture(j)=(Mid(strDat, 5))
      j=j+1
    Case "提出先=": strTramsPath=Mid(strDat, 5)
    Case "参照元=": strReferencePath=Mid(strDat, 5)
    Case "[課題提]": blnFlg=True
    Case "": blnFlg=False
    Case Else:
  End Select
Loop
Close#intFileNo
End Sub
Function getNumber()
  getNumber=strNumber
End Function
Function getLecture()
  getLecture=strLecture
End Function
Function getSoftName()
  getSoftName=strSoftName
End Function
Function getKadaiName()
  getKadaiName=strKadaiName
End Function
Function getTramsPath()
  getTramsPath=strTramsPath
End Function
Function getReferencePath()
  getReferencePath=strReferencePath
End Function

```

(6) 関数「ComCtl32Bas」のプログラムコード⁽⁶⁾

[参照] ボタンを押して特定ファイルの格納場所とそのファイル名を選択し、ダイアログ「ファ

イルを開く」を表示するのに必要な API 宣言, 定数, 関数を一つにまとめたモジュールである。

```

**** 定数定義部分および構造体定義部分は省略 ****
Public Function Select_OpenFileName (ByVal hWnd As Long,
                                     Optional Title As Variant,
                                     Optional InitDir As Variant,
                                     Optional Exp As Variant) As String

    Dim OF As OpenFileName
    Dim tmp As String
    Dim stu As Long
    Dim filter As String
    Dim strfilter As String
    If Exp = "*" Then
        strfilter = "全て (*.*)"
    Else
        strfilter = Right(Exp, 3) & "ファイル (" & Exp & ")"
    End If
    tmp = String$(5120, vbNullChar)
    filter = "全て (*.*)" & vbNullChar & "*" & vbNullChar & vbNullChar
    'filter = "sei ファイル (*.sei)" & vbNullChar
    filter = strfilter & vbNullChar
    If Not IsMissing(Exp) Then
        filter = filter & Exp
    End If
    With OF
        .lStructSize = Len(OF)           '構造体長
        .hwndOwner = hWnd               'Owner ハンドル
        .lpstrFile = tmp                 '戻り値バッファ
        .nMaxFile = 5120                 '戻り値バッファ長
        .lpstrFilter = filter
        .nFilterIndex = 1
        .Flags = OFN_FILEMUSTEXIST Or OFN_HIDEREADONLY Or OFN_PATHMUSTEXIST
        If Not IsMissing(Title) Then .lpstrTitle = Title      'ダイアログタイトル
        If Not IsMissing(InitDir) Then .lpstrInitialDir = InitDir '初期ディレクトリ
    End With
    stu = GetOpenFileName(OF)
    If stu Then
        Select_OpenFileName = Left$(OF.lpstrFile, InStr(OF.lpstrFile, vbNullChar) - 1)
    End If
End Function

```

4.2 受講者名簿更新処理 (Access VBA)

この処理は、出席票と課題ファイルの2種類を含む「提出物全体」フォルダから出席票のみをピックアップし、その中にある学籍番号と講義名を抽出して受講者名簿に登録を行う。

(1) 第7図画面bの「参照」ボタンを押して開始される処理

[参照] ボタンにより提出物全体フォルダを選定する作業を通じてフォルダの所在を示すパスを確定する処理である。

```
Private Sub btnReference_Click()
    Dim strTtl As String, strDest As String, lngRet As Long
    strTtl="フォルダを選択してください。"
    Do
        lngRet=Browse_ForFolder(strDest, , strTtl, , Me.hWnd)
        If lngRet=False Then
            Exit Do
        ElseIf strDest="" Then
            If MsgBox ("フォルダを選択してください。処理を中止する場合はキャンセルをクリックしてください。", vbInformation+vbOKCancel)<>vbOK Then
                lngRet=False
            Exit Do
            End If
        End If
        Loop Until lngRet And strDest>""
        If strDest>"" Then
            Me.txtFilePath=strDest          '提出物全体フォルダまでのパスを示す
        End If
    End Sub
```

(2) 第7図画面bの「開始」ボタンを押して行われる処理 (リスト中の四角で囲った部分は、関数名またはサブプロシージャ名であることを示す)

```
Private Sub btnStart_Click()
    Dim db As DAO.Database, rcdStudent As DAO.Recordset, rcdChange As DAO.Recordset
    Dim strFilePath As String, strFileName As String
    Dim strNumber As String, strLecture As String
    Dim strdmy As String, i As Integer
    Dim intCnt As Integer, j As Integer
    '提出物全体フォルダを実際に検索することの可能性を確認
    With Application.FileSearch
```

```

.NewSearch
.LookIn=strFilePath
.SearchSubFolders=False
.FileName="*.*"
.MatchAllWordForms=True
If vbCancel=MsgBox("検索結果はこれでいいですか" & Chr$(13)&Chr$(13)&_
    "検索場所:" & strFilePath, vbOKCancel)Then
    Exit Sub
End If
Set objFs=CreateObject("Scripting.FileSystemObject")    'パスのチェック
If objFs.FolderExists(strFilePath)=False Then
    MsgBox("指定したパスは存在しません")
    Exit Sub
End If
If Right(strFilePath, 1)="¥" Then    'ファイル名抽出の準備
    sngSearchLen=Len(strFilePath)+1
Else
    sngSearchLen = Len (strFilePath) + 2
End If

'取り込んだ個々の提出ファイル名を判定し、出席票のみを抽出
'学籍番号が未登録ならば学籍番号と講義名の双方を登録、登録済みならば講義名のみを登録

If .Execute(>0)Then
Set db=OpenDatabase (CurrentDb.Name)    'データベースと各テーブルの設定
Set rcdStudent=db.OpenRecordset("受講者名簿", dbOpenDynaset)
Form_prgBar.SetFocus    '処理バーの設定
For i=1 To .FoundFiles.Count
    strdmy=Mid(.FoundFiles(i), sngSearchLen)
    strFileName = Char_Modification (rcdChange, strdmy)
    If [Table_Judg] (strFileName, j)=True Then    '出席票の学籍番号と講義名を抽出
        strNumber=Left(strFileName, 6)    '学籍番号を代入
        strLecture=Mid(strFileName, 7, 2)    '講義名を代入
        [AdditionStudentTable] rcdStudent, strNumber, "", strLecture
        intCnt=intCnt+1
    End If
    Form_prgBar.Progress.Value=Int(i / .FoundFiles.Count * 100)    '処理バーの更新
    DoEvents
Next
DoCmd.Close acForm, Form_prgBar.Name, acSaveNo
Else
    MsgBox "検索条件を満たすファイルはありません。"
End If

```

```

End With
MsgBox intCnt & "件のデータを受講者名簿テーブルに記録しました"
DoCmd.Close acForm, Me.Name, acSaveNo
End Sub

```

(3) 関数「Table_Judg」のプログラムコード

出席票か課題ファイルかを判定し、その結果を他のプログラムに引き渡す処理

```

Function Table_Judg (strFileName As String, ByRef i As Integer)
    If Len(strFileName)>=10 Then
        If IsNumeric(Mid(strFileName, 7+i, 3))Then 'ファイル名から日付を抽出するための調整
            Table_Judg=True
        Else
            Table_Judg=False
        End If
    Else
        Table_Judg=False
    End If
End Function

```

(4) サブプロシージャ「AdditionStudentTable」のプログラムコード

学籍番号が名簿に未登録ならば、学籍番号と講義名を新たに登録し、学籍番号が名簿に登録されていて講義名が未登録ならば、新たに講義名を登録する処理

```

Option Compare Database
Sub AdditionStudentTable(rcdStudent As DAO.Recordset, strNumber As String, strName As String, strLecture As String)
    Dim strSearchSQL As String
    strSearchSQL="学籍番号="& strNumber &""
    With rcdStudent
        If .RecordCount<>0 Then
            .MoveLast
        End If
        .FindFirst strSearchSQL
        If .NoMatch Then '新たな学籍番号なのでテーブルに登録
            .AddNew
            !学籍番号=strNumber
            !講義名 1=strLecture
            If strName<>"" Then

```



```

!氏名=strName
End If
.Update
.Bookmark=.LastModified
Else
'学籍番号は既にテーブルに登録済
If(!講義名 1=strLecture)Or(!講義名 2=strLecture)Or(!講義名 3=strLecture)Then
'lngErrCnt=lngErrCnt+1
Else
.Edit
If IsNull(!講義名 1)Then
'講義名 1 フィールドに登録
ElseIf IsNull(!講義名 2)Then
'講義名 2 フィールドに登録
ElseIf IsNull(!講義名 3)Then
'講義名 3 フィールドに登録
Else
'lngErrCnt=lngErrCnt+1
End If
.Update
.Bookmark=.LastModified
End If
End If
End With
End Sub

```

4.3 年間提出物集計処理 (Access VBA)

これは、年度初めからプールしてある提出物を判定し、データベースの「出席」テーブル、「課題」テーブルおよび「エラー」テーブルに格納する中核的な処理である。

(1) サブプロシージャ「getAllFile1」のプログラムコード

提出された全部のファイルをデータベース内のテーブルに格納する処理

```

Sub getAllFile1(strSearchChar As String, strSearchPath As String)
Dim DbA As DBAccess
Dim i As Integer, intFlg As Integer
Dim intErrCnt As Integer, str As String, strdmy As String
Dim sngSearchLen As Single, objFs As Object
intErrCnt=0
'規格外データ件数カウンタをゼロクリア
If strSearchChar="" Then
strSearchChar="*.*"

```

```

End If
If strSearchPath="" Then
    strSearchPath="j:¥情報技術¥課題提出¥2004 全部¥"
End If
With Application.FileSearch                                '検索条件の設定
    .NewSearch
    .LookIn=strSearchPath
    .SearchSubFolders=False
    .FileName=strSearchChar
    .MatchAllWordForms=True
    .FileType=msoFileTypeAllFiles
    intFlg=MsgBox("検索条件はこれでいいですか" & Chr$(13)& Chr$(13)&_
        "検索文字:" & strSearchChar & Chr$(13)& _
        "検索場所:" & strSearchPath, vbOKCancel)
    If intFlg<>1 Then
        Exit Sub
    End If
    Set objFs=CreateObject("Scripting.FileSystemObject")    'パスのチェック
    If objFs.FolderExists(strSearchPath)=False Then
        MsgBox("指定したパスは存在しません")
        Exit Sub
    End If
    If Right (strSearchPath, 1)="¥" Then                      'ファイル名抽出の準備
        sngSearchLen=Len(strSearchPath)+1
    Else
        sngSearchLen=Len(strSearchPath)+2
    End If
    If .Execute()>0 Then                                     '検索開始
        Form_prgBar.SetFocus
        MsgBox .FoundFiles.Count & " 個のファイルが見つかりました。" '第9図画面 d
        Set Dbա.dbs=OpenDatabase(CurrentDb.Name, False)
        Set Dbա.ErrRst=Dbա.dbs.OpenRecordset("ERR_TABLE", dbOpenDynaset)
        Set Dbա.KadaiRst=Dbա.dbs.OpenRecordset("課題", dbOpenDynaset)
        Set Dbա.AttendanceRst=Dbա.dbs.OpenRecordset("出席", dbOpenDynaset)
        For i=1 To .FoundFiles.Count
            intFlg=Insert_Table(str, Dbա)                    'テーブルにデータを追加
            'テーブルにデータを追加出来ないときにそのファイル名を ERR_TABLE に追加する
            If intFlg<>0 And strdmmmy<>"" Then
                With Dbա.ErrRst
                    .AddNew
                    !ファイル名=strdmmmy
                    !エラー原因=intFlg
                    .Update
                End With
            End If
        Next i
    End If
End With

```

```

        .Bookmark=.LastModified
    End With
    intErrCnt=intErrCnt+1
End If
Form_prgBar.Progress.Value=Int(i/.FoundFiles.Count*100)
DoEvents
Next i
DoCmd.Close acForm, Form_prgBar.Name, acSaveNo
Dbա.ErrRst.Close
Dbա.KadaiRst.Close
Dbա.AttendanceRst.Close
Dbա.ChangeRst.Close
Dbա.dbs.Close
Else
    MsgBox "検索条件を満たすファイルはありません。"
End If
End With
If intErrCnt>0 Then '追加出来なかったファイルの表示
    MsgBox intErrCnt & "個のファイルを追加できませんでした。"& vbNewLine_
        & "追加できなかったファイルは ERR_TABLE に記録してあります。"
End If
End Sub

```

(2) 関数「Insert_Table」のプログラムコード

個々の提出ファイルを格納する処理

```

Function Insert_Table(strFileName As String, ByRef Dbա As DBAccess)
    Dim Check_Record As DAO.Recordset
    Dim strStudentNmb As String, strRecordCheck As String
    Dim strLectureName As String
    Dim strDay As String, strKadaiName As String
    Dim dummy As String, i As Integer
    Set Check_Record=Dbա.dbs.OpenRecordset("受講者名簿", dbOpenSnapshot, dbReadOnly)
    strStudentNmb=Left(strFileName, 6)
        '入力されたデータの学籍番号が受講者名簿テーブルにあるか否かを検索
    strRecordCheck="学籍番号="& strStudentNmb & ""
    With Check_Record
        .MoveLast
        .FindFirst strRecordCheck
        If .NoMatch Then
            Insert_Table=1
        End If
    End With
End Function

```

```

Exit Function
End If
End With
Check_Record.Close

                                '開くテーブル名をファイル名から判定
If [Table_Judg](strFileName, i)=False Then      '課題の場合
Set Check_Record=DbA.dbs.OpenRecordset("課題", dbOpenSnapshot, dbReadOnly)
strKadaiName=Mid(strFileName, 7)

                                'チェックプログラム
strRecordCheck="学籍番号=" & strStudentNmb & " AND 課題名=" & strKadaiName & ""
With Check_Record
.FindFirst strRecordCheck
If .NoMatch Then
[Add_Data] DbA.KadaiRst, strStudentNmb, strKadaiName, dummy, strFileName, False, i
Else
Insert_Table=2
Exit Function
End If
End With
Else                                '出席票の場合
Set Check_Record=DbA.dbs.OpenRecordset("出席", dbOpenSnapshot, dbReadOnly)
strLectureName=Mid (strFileName, 7, i)
strDay=Mid(strFileName, 7+i, 6)
strRecordCheck="学籍番号=" & strStudentNmb & " AND 講義名=" & strLectureName
& " AND " & "_&"日付=" & strDay & ""
With Check_Record
.FindFirst strRecordCheck
If .NoMatch Then
[Add_Data] DbA.AttendanceRst, strStudentNmb, strLectureName, strDay, strFile
Name, True, i
Else
Insert_Table=3
Exit Function
End If
End With
End If
Check_Record.Close
Insert_Table=0
End Function

```

(3) 関数「Add_Data」のプログラムコード

提出物のファイル名を「出席票」テーブルか「課題」テーブルのどちらかに格納

```

Function Add_Data(rst 1 As DAO.Recordset, strStudentNmb As String, strLectureName As
String, strDay As String, strFileName As String, blnJudg As Boolean, i As Integer)
  If blnJudg=False Then
    With rst 1
      .AddNew                                '課題の場合
      !学籍番号=strStudentNmb
      !課題名=strLectureName
      !ファイル名=strFileName
      .Update
      .Bookmark=.LastModified
    End With
  Else
    With rst 1
      .AddNew                                '出席票の場合
      !学籍番号=strStudentNmb
      !講義名=strLectureName
      !日付=strDay
      !ファイル名=strFileName
      .Update
      .Bookmark=.LastModified
    End With
  End If
End Function

```

4.4 年間提出物提示処理 (VB)

これは、データベース内の「出席」テーブルと「課題」テーブルに格納されている出席状況と課題提出に関するデータを取り出して学生のパソコン画面に提示する処理である。

(1) 関数「GetPastData」のプログラムコード

学生データのレコードセットを取得

```

Public Function GetPastData() As udtStudentInfo
  On Error GoTo ErrorHandler:
  Dim typStudentInfo As udtStudentInfo
  Dim strPath          As String          **** データベースまでのパスを取得
                                          ****学籍番号を取得(環境情報より)
  typStudentInfo.strNumber=modParameter.getNumber
                                          ****コネクションを張る
  mod100Database.gConnect modParameter.GetDBFullPath
  GetStudentInfo typStudentInf

```

```

**** 出席データを取得
GetAttendData typStudentInfo.typAttendInfo, typStudentInfo.strNumber
**** 課題データを取得
GetKadaiData typStudentInfo.strKadai, typStudentInfo.strNumber
**** 拡張子ごとの課題提出数を取得

GetKadaiCount typStudentInfo
mod100Database.gDisconnect
GetPastData=typStudentInfo
Exit Function
ErrorHandler:
                                     'エラー時の処理
    mod100Database.gDisconnect
    Err.Raise ERROR_SYSTEM
End Function

```

(2) 関数「GetStudentInfo」のプログラムコード

学生データの取得（同一学籍番号をもつレコードセットを取得する SQL を生成）

```

Private Function GetStudentInfo(ByRef vtypStudentInfo As udtStudentInfo)
On Error GoTo ErrorHandler:
    Dim objRec As ADODB.Recordset
    Dim strSQL As String
    Dim lngIdx As Long
    strSQL=""
    strSQL=strSQL & "select 学籍番号.氏名 as name "
    strSQL=strSQL & "from " & mod 000 CommonConstants.TM_STUDENT & "
    strSQL=strSQL & "where 学籍番号.学籍番号 = " & vtypStudentInfo.strNumber & " "
                                     'レコードセットを取得
    Set objRec=mod100Database.gCreateRs(strSQL, adOpenStatic, adLockReadOnly)

```

(3) 関数「GetAttendData」のプログラムコード

学生データの取得（講義別出席回数をカウントする SQL を生成）

```

Private Function GetAttendData(ByRef vtypAttendInfo() As udtAttendInfo, _
                                ByVal vstrNumber As String)
On Error GoTo ErrorHandler:
    Dim objRec As ADODB.Recordset
    Dim strSQL As String
    Dim lngIdx As Long

```

```

strSQL=""
strSQL=strSQL & "select 講義名,"
strSQL=strSQL & "count (講義名) as attendance "
strSQL=strSQL & "from " & mod000CommonConstants.TT_ATTEND & " "
strSQL=strSQL & "where 学籍番号=" & vstrNumber & " "
strSQL=strSQL & "GROUP BY 講義名"
                                'レコードセット取得
Set objRec=mod100Database.gCreateRs (strSQL, adOpenStatic, adLockReadOnly)
lngIdx=0
ReDim vtypAttendInfo(lngIdx)
With objRec                                'レコードセットのデータを構造体に格納
    Do Until .EOF
        If IsNull(.Fields("講義名"))=False Then '値が NULL の場合は処理しない
            ReDim Preserve vtypAttendInfo(lngIdx)
            vtypAttendInfo(lngIdx).strLectureName=Trim$(.Fields("講義名").Value)
            vtypAttendInfo(lngIdx).lngAttendance=.Fields("attendance").Value
            lngIdx=lngIdx+1
        End If
        .MoveNext
    Loop
    .Close
End With
Set objRec=Nothing                        'レコードセットオブジェクト解放
Exit Function
ErrorHandler:
    Set objRec=Nothing                    'レコードセットオブジェクト解放
    Err.Raise ERROR_SYSTEM
End Function

```

(4) 関数「GetKadaiData」のプログラムコード

学生データの取得（課題名を取得する SQL を生成）

```

Private Function GetKadaiData(ByRef vstrKadai() As String, _
                                ByVal vstrNumber As String)
On Error GoTo ErrorHandler
    Dim objRec As ADODB.Recordset
    Dim strSQL As String
    Dim lngIdx As Long
    strSQL=""
    strSQL=strSQL & "select 課題名 "
    strSQL=strSQL & "from " & mod000CommonConstants.TT_KADAI & " "

```

```

strSQL=strSQL & "where 学籍番号=" & vstrNumber & ""
                                'レコードセット取得
Set objRec=mod100Database.gCreateRs(strSQL, adOpenStatic, adLockReadOnly)
lngIdx=0
ReDim vstrKadai(lngIdx)
With objRec
  Do Until .EOF
    If IsNull(.Fields("課題名"))=False Then '値が NULL ならば処理しない
      ReDim Preserve vstrKadai(lngIdx)
      vstrKadai (lngIdx)=Trim$(.Fields("課題名").Value)
      lngIdx=lngIdx+1
    End If
    .MoveNext
  Loop
  .Close
End With
Set objRec=Nothing 'レコードセットオブジェクト解放
Exit Function
ErrorHandler:
Set objRec=Nothing 'レコードセットオブジェクト解放
Err.Raise ERROR_SYSTEM
End Function

```

(5) 関数「GetKadaiCount」のプログラムコード

拡張子別の課題提出数を集計

```

Private Function GetKadaiCount(ByRef vtypStudentInfo As udtStudentInfo)
On Error GoTo ErrorHandler:
  Dim strExp As String '取得した課題名の拡張子
  Dim varExp As Variant '集計するソフト種類
  Dim varSoft As Variant 'ソフト名
  Dim lngCnt() As Long
  Dim lngIdx As Long
  Dim lngCntIdx As Long
  Dim lngSum As Long '*** 提出課題合計
  varExp=Array("BMP", "DOC", "XLS", "PPT", "HTML", "MDB", "ER 1", "JAVA", "CPP", "VBP")
  varSoft=Array("ペイント", "Word", "Excel", "Power Point", "HTML", "Access", "DB 設計",
"JAVA", "C,C++", "VB")
  ReDim lngCnt(UBound(varExp))
  With vtypStudentInfo '拡張子ごとの提出数の集計
    For lngIdx=LBound(.strKadai) To UBound(.strKadai)

```



```

'拡張子を抽出し大文字に変換
strExp=Mid$(strKadai(lngIdx), InStrRev(strKadai(lngIdx), ".", vbBinaryCom-
pare)+1)
strExp=UCase(strExp)
For lngCntIdx=LBound(varExp)To UBound(varExp)
    If StrComp(strExp, varExp(lngCntIdx), vbBinaryCompare)=0 Then
        lngCnt(lngCntIdx)=lngCnt(lngCntIdx)+1
    Exit For
End If
Next lngCntIdx
Next lngIdx
End With
With vtypStudentInfo                                '集計した値を構造体に格納
    For lngIdx=LBound(varSoft)To UBound(varSoft)
        ReDim Preserve .typKadaiInfo(lngIdx)
        .typKadaiInfo(lngIdx).strKadaiName=varSoft(lngIdx)
        .typKadaiInfo(lngIdx).lngKadai=lngCnt(lngIdx)
    Next lngIdx
End With
With vtypStudentInfo                                '「その他」の算出
    lngSum=0
    'リストボックスに表示されてる値を合計(ペイント, Word, ...,C++,VB)
    For lngIdx=LBound(.typKadaiInfo)To UBound(.typKadaiInfo)
        .lngKadaiSum=.lngKadaiSum+.typKadaiInfo(lngIdx).lngKadai
    Next lngIdx
    '現在提出されている課題の合計を取得
    If Len(Trim$(strKadai(LBound(strKadai))))=0 Then
        '課題がひとつも提出されていない場合
        lngSum=0
    Else
        lngSum=UBound(strKadai)+1    '課題が提出されている場合
    End If
    ReDim Preserve .typKadaiInfo(UBound(.typKadaiInfo)+1)
    .typKadaiInfo(UBound(.typKadaiInfo)).strKadaiName="その他"
    .typKadaiInfo(UBound(.typKadaiInfo)).lngKadai=lngSum-.lngKadaiSum
End With
Exit Function
ErrorHandler:
    Err.Raise ERROR_SYSTEM
End Function

```

(6) サブプロシージャ「SetFormData」のプログラムコード

提出状況を格納した構造体の学生情報をユーザのパソコン画面に表示

```

Private Sub SetFormData(ByRef vtypStudentInfo As udtStudentInfo)
On Error GoTo ErrorHandler
    Dim lvwAttendItem As ListItem
    Dim lvwKadaiItem As ListItem
    Dim lngIdx As Long
    With vtypStudentInfo
        txtNumber.Text = .strNumber
        txtName.Text = .strName
        For lngIdx = LBound(.strKadai) To UBound(.strKadai) **** 提出課題名データを提示
            lstKadai.AddItem .strKadai(lngIdx)
        Next lngIdx
        lvwKadaiCnt.View = lvwReport **** 課題提出数を設定
        For lngIdx = LBound(.typKadaiInfo) To UBound(.typKadaiInfo)
            Set lvwKadaiItem = lvwKadaiCnt.ListItems.Add
            lvwKadaiItem.SubItems(1) = .typKadaiInfo(lngIdx).strKadaiName
            lvwKadaiItem.SubItems(2) = CStr(.typKadaiInfo(lngIdx).lngKadai)
        Next lngIdx
        Set lvwKadaiItem = Nothing
        lvwAttend.View = lvwReport **** 出席状況データを設定
        For lngIdx = LBound(.typAttendInfo) To UBound(.typAttendInfo)
            If Len(Trim$(.typAttendInfo(lngIdx).strLectureName)) > 0 Then
                Set lvwAttendItem = lvwAttend.ListItems.Add
                lvwAttendItem.SubItems(1) = .typAttendInfo(lngIdx).strLectureName
                lvwAttendItem.SubItems(2) = CStr(.typAttendInfo(lngIdx).lngAttendance)
            End If
        Next lngIdx
        Set lvwAttendItem = Nothing
        txtSum.Text = CStr(.lngKadaiSum)
    End With
Exit Sub
ErrorHandler:
    'エラー時の処理
    Set lvwKadaiItem = Nothing
    Set lvwAttendItem = Nothing
    MsgBox Err.Description
    Err.Raise ERROR_SYSTEM
End Sub

```

おわりに

本稿で検討した実習支援システムの開発（VBバージョン6とAccess 2000）には2年の歳月と多大な労力を費やしたが、その完成後も小規模な機能強化を継続しているので、現在も円滑に稼働している。そしてその適用によって受講生の課題作成意欲の高まりは当初の予想を越え、多くの受講生がITのスキルアップに向かったことを幸いに思っている。今後予定されているパソコンシステムのグレードアップ時（2005年夏）には、新バージョンであるVB.netとAccess 2003によりシステムの再構成を図る計画であるが、その際に「提出ファイルの自動採点機能」や「提出課題ファイルの得点表示」等の新機能を追加する予定である。

《注》

- (1) データベースとは、複数の業務での使用を目的として多種類のデータ群を一箇所に集積した統合ファイルである。それは、データを簡単かつ効率よく管理する手段であり、企業における社員データや顧客データの一元管理（データ重複の排除）などに利用されているが、最近では家庭においても住所録や家計簿の管理にデータベースが使用されている。全社的な情報システムの統合は、大小多数のデータベースを連係することによって達成されている。データベースの種類には、データの関係を木構造の親子関係（1対多）で表現する階層モデル、（多対多の）親子関係で表現するネットワークモデル、2次元表で表現する関係モデル、データに操作をカプセル（内包）化したオブジェクト指向モデルがある。その中で関係（リレーショナル）データベースは、エンドユーザがパソコン上で使用するレベルから企業全体を網羅する大規模データベースまで広い範囲で使用されている。
- (2) データベース設計（データモデリング）とは、実世界での業務の仕組をコンピュータ（特にデータベース）に反映させるための手法である。それは、業務を構成するエンティティとその関係を認識することにより業務に関わる情報の構造を定義し、ビジネスルールをとらえて情報システムを定義するプロセスである。情報システム開発の手法であるデータ中心設計においては、データをプロセスに付属させる視点より全体的・業務横断的で共有化すべき主要なエンティティの確定を上位におき、その下位レベルとして個別業務に関する全エンティティを詳細に定義するが、データモデリングツールは、その方法論にもとづいてデータベース設計を支援するための種々の機能をもつ。また業務設計は、実世界の仕事の仕組を可視化しビジネスプロセスとして把握し、プロセスの全体像を明確化することによりその分析や改善を行うことを可能にするツールである。
- (3) 本稿作成の分担は、Aが全体を統括し、B（卒業後に国内ソフト企業にSEとして勤務）は卒業論文制作の一環として実習支援ソフトの開発を行った。本稿に示すプログラムコードは、Bによる作成である。
- (4) コンピュータ実習態勢の基本要件に関しては、下記の3編の論文で考察している。
 - ① 加藤武信、尾辻斗志：コンピュータ実習教育における指導員制の導入とその拡充過程—文科学部における事例に基づく考察—、城西情報科学研究創刊号、pp.5～19, 1989.
 - ② 加藤武信、尾辻斗志：プログラミング学習における知識獲得の場に関する考察、城西経済学会誌、第24巻第2・3号、pp.1～31, 1990.
 - ③ 加藤武信：プログラミング実習教育の基本要件に関する考察、平成2年度情報処理教育研究集

会報告書（文部省・京都大学共催），pp.238～244, 1990.

- (5) 加藤武信, 境野大地, 水野政友：コンピュータ活用教育における基本要件の拡充とその評価, 平成16年度情報処理教育研究集会報告書（文部科学省・名古屋大学共催）, pp.240～243, 2004.
- (6) このソースファイルは, <http://www.loadsystem.net/api/api01.htm> からダウンロードしたものである。

(2005.2完)

《Summary》

Some Considerations of Design and Practical Use of
a Database for a Computer Training Support

By Takenobu KATO
Masatomo MIZUNO

We have developed a computer training support system of programming language VB (Visual Basic) and relational database Access. This system electronically receives presentation things (e. g. attendance votes and exercises) from students, and informs them of these data soon after gathering them. As a result of having used this system in a computer course, it created the following effects: increased students' motivation for study, more exercises submitted than before, promoted efficiency of a course management.

This paper is intended to introduce the results analysis that applies this support system. Furthermore, we are going to explain mechanism of dataprocessing of this system and show the program code, including database programming skills about some main subsystems.