

COBOL プログラミングを習得した生徒が、『ソフトウェア開発』の演習において、イベント駆動型 Basic の基本的な内容を学習するためのテキスト試案作成に関する研究

広島市立広島商業高等学校教諭 中 堀 恵

研究主題設定の理由

本校情報処理科では、表1に示したよう全国商業高等学校協会主催(以下、「全商」)の情報処理検定1級プログラミング部門と国家試験である情報処理技術者試験の基本情報技術者の資格取得のための内容を中心に汎用コンピュータを用いてCOBOLプログラミングを第1学年及び第2学年で指導し、第3学年では1年間を通してソフトウェア開発の演習を行っている。

ソフトウェア開発の演習では、経営情報科目の総まとめとして、COBOLを中心に、生徒の興味と希望などに応じてExcel VBAやC、Javaなどでの開発も指導してきた。COBOL以外での開発を指導する場合、他の言語を指導するための教材が十分に用意できていないため、市販のテキストを活用しているが、市販のテキストは、基礎から学ぶものか、一定のスキルがあるものを前提としたものである。したがって、生徒の開発では必要のない内容も多く、また、COBOLプログラミングの知識・技能が十分に活かすことができるような記述になっていないため、言語の習得に多くの時間を費やしてしまい、生徒が基本計画段階で考えたような作品を作り上げることが、時間的に困難であった。

また、社会の求人ニーズは、日本商工会議所の「総合的人材ニーズ調査」(平成12年)や福山平成大学の「大学における情報処理教育に関するアンケート調査」(平成13年)などによると、情報処理関連の職種や新入社員に対して、資格としては経済産業省の情報処理技術者を求め、プログラミングの技能としてはVisual Basic(以下、「VB」)、COBOL、C、Javaを求めている。

これらの状況を踏まえ、生徒の興味・関心や社会のニーズに対応するため、これまでどおり、資格取得に向けてCOBOLプログラミングを指導し、COBOLプログラミングを習得した生徒には、発展的な学習として、VBをソフトウェア開発の演習の中で指導することが効果的であると考えた。そして、指導にあたっては、既習のCOBOLプログラミングの知識・技能を生かしたテキストを作成する必要がある、教師にとってもこのようなテキストを活用することが、支援・指導に有効であると考え、テキスト試案の作成を行うこととした。

研究の方法

研究の目的を達成するための基礎的研究として、VBの最新バージョンであるVisual Basic.NET(以

表1 本校情報処理科の経営情報科目群の指導内容

学年	科目 (H14入学生まで)	内 容	科目 (H15入学生より)	内 容
1年	情報処理 プログラミング	COBOLプログラミング(中心)、情報処理検定2級(プログラミング部門)、Excelの利用ほか	情報処理 プログラミング	基本的には変更なし
2年	プログラミング 情報管理 経営情報	COBOLプログラミング(中心)、情報処理検定1級(プログラミング部門)2級(ビジネス情報部門)、情報処理技術者試験(基本情報)、Excelの利用(VBA含む)ほか	プログラミング ビジネス情報 文書デザイン	情報処理検定1級(プログラミング部門またはビジネス情報部門)、情報処理技術者試験(基本情報または初級シスアド)
3年	課題研究	COBOLによるソフトウェアの開発(一部Excel VBAほか)	(総合実践)	簿記会計に関する実践、経営情報に関する実践(ソフトウェア開発)

下、「VB.NET」)について調べ、COBOL と対比させて特徴を整理するとともに、ネットワーク環境で実習するための設定について調査し、設定を変更する。

ソフトウェア開発の演習に必要かつ、基本的な内容について、量的なことも考えながら整理し、テキスト試案を作成する。また、生徒に作成したテキストを試用させることを通して、改善のための情報を収集し、テキストの改善に役立てる。

研究の内容

1 研究主題に関する基礎的研究

テキストを作成するにあたって、VB の最も新しいバージョンである VB.NET について調査し、VB の特徴を整理するとともに、ネットワーク環境での実習のために必要な設定について確認した。

(1) VB.NET

Microsoft 社は、スタンドアロンでもネットワークでも利用できるアプリケーションを開発、提供、運用できるソフトウェアを提供している。それが、Visual Studio.NET と Microsoft.NET Framework (以下、「.NET Framework」)である。Visual Studio.NET は、VB.NET, Visual C++.NET, Visual C#.NET, J Script といったプログラム言語が使用できる開発ツールである。.NET Framework は、Visual Studio.NET で作成されたアプリケーションを実行するためのソフトウェアで、アプリケーションと OS の間に存在する。

Visual Studio.NET に組み込まれている VB.NET は、次の 3 タイプのプログラムを作成できる。

Windows アプリケーション
 スタンドアロンで動作する最も基本的なアプリケーション
 Web アプリケーション
 インターネット技術を利用し、ネットワーク環境で動作する Web ページ用のアプリケーション
 XML Web サービス
 Web ページと通信し、サーバー上で処理を行うプログラム

また、Excel や Access との連携も容易にできる。

以前のバージョンである VB Version6 (以下 VB6)

は、対応する OS である Windows に特化していた。VB6 で開発したアプリケーションは、Windows の動作するコンピュータ用の機械語に変換し、実行していた。これに対して VB.NET は、中間コード (MS IL) に変換され、実行時に逐次解釈実行を行う。したがって、MSIL を解釈実行するプログラムがあれば、Windows 以外の環境でも動作させることができる。

(2) VB でのプログラミング

ア プログラミングの手順

COBOL と VB のプログラミング手順を対応させて整理すると、図 1 のようになる。

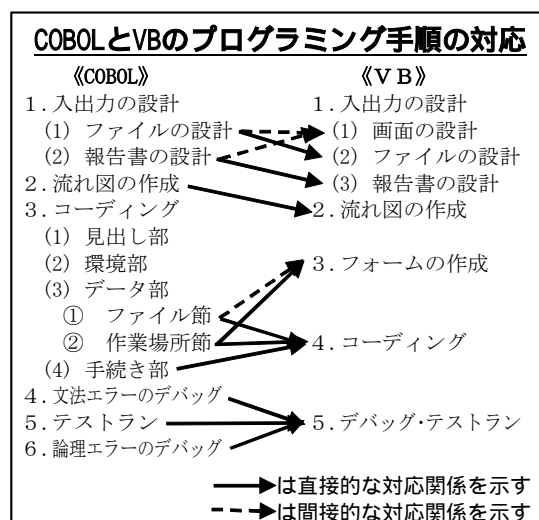


図 1 COBOL と VB のプログラム手順の対応

VB は、対話型のプログラムの作成に適した言語であるので、入出力設計は画面の設計が中心となり、必要に応じてファイルの設計、報告書の設計を行うことになる。流れ図の作成は、基本的に COBOL と同様に構造化プログラミングの技法で行うことができる。命令や処理のさせ方に若干の違いはあるが、コーディングの際に修正すればよい。フォームの作成は、画面の設計にしたがい行う。フォームは、プログラムの土台となるウィンドウである。ここに、データの入出力をするためのテキストボックスなどを貼り付け、入出力の項目を定義することになる。なお、表示形式については、コーディングの中でも行える。コーディングは、COBOL では処理の記述をする前に、項目の定義を行うが、VB ではフォー

ムの中に定義された項目以外は、いつでも定義を行える。処理については、流れ図にしたがって**VB**の命令に置き換え、記述していく。デバッグ、テストランについては、対話型のデバッガが用意されているので、テストランをしながらデバッグを行うことができる。

このように、**COBOL**と**VB**では、基本的な手順に大きな違いはないが、対話型のプログラムを作成することによる違いが、多少存在する。

イ ファイルの処理

COBOLと**VB**のファイル処理の対応を整理すると、図2のようになる。

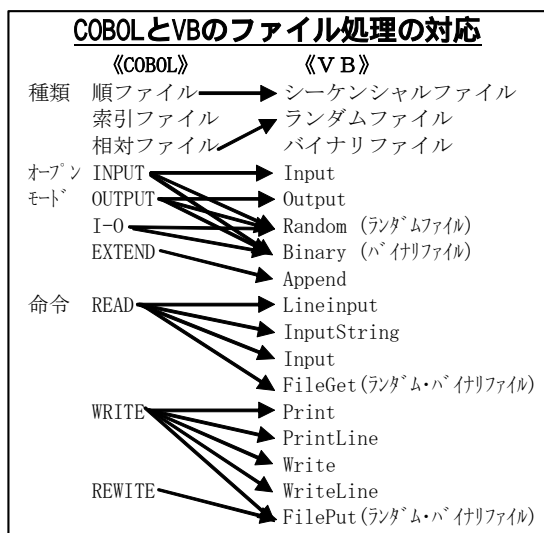


図2 COBOLとVBのファイル処理の対応

ファイルの種類は、**COBOL**での順ファイルがシーケンシャルファイルに、相対ファイルがランダムファイルにあたる。**VB**ではその他にバイナリファイルがあり、**COBOL**の索引ファイルにあたるものがない。オープンモードは、シーケンシャルファイルについては**COBOL**と同様で、ランダムファイルとバイナリファイルでは、ファイルの種類を指定してオープンする。入出力命令は、**COBOL**の**READ**、**WRITE**に対応する命令が、シーケンシャルファイルでは、いくつにも細かく分かれる。ランダムファイルとバイナリファイルの命令は、入力、出力一つずつになっている。

VBでは、最も汎用的に用いられるシーケンシャルファイルを扱う命令が多く、生徒がすべてを理解

し使い分けることは、容易ではないように思われる。
ウ 印刷処理と**Excel**の利用

バッチ型の**COBOL**のプログラムから印刷するような文字列のみの印刷であれば、**VB**からでも比較的容易に印刷できる。しかし、罫線を入れたり、フォントの種類やサイズを指定したりするには、印刷の際に位置や文字列編集などさまざまな指定をする必要があり、思うような印刷をすることは容易でない。ところが、**VB**では、**Excel**の機能呼び出して利用することが可能であり、**Excel**を利用することにより、罫線を引いたり、フォントの種類やサイズを自由に設定したりできるので、帳票印刷が容易にできる。また、**Excel**を利用すれば、索引ファイルやデータベースと同様な処理も実現できる。

VBでプログラムを作成する場合は、**Excel**を印刷などにおいて、活用することが有効である。

エ 例外処理

COBOLのバッチ型のプログラムは、入力データをあらかじめチェックし、エラーを修正し、エラーのないデータを処理する。**VB**の対話型のプログラムは、エラーデータが入力される可能性が非常に高いので、あらゆるケースを想定し、例外処理を準備しておく必要がある。そのため、**VB**では例外処理を記述するための構文(**Try~Catch~Finally~**)が用意されている。

(3) **VB**で作成する流れ図の基本形

プログラムを作成するには、入力形式と出力形式を決めた後に作成する流れ図が非常に重要である。流れ図では、処理手順を順序立てて整理し、決められた記号で表す。流れ図ができれば、あとはプログラム言語に置き換える作業のみとなる。ただし、プログラム言語による多少のプログラミングテクニックを要する場合はある。

生徒は、**COBOL**プログラミングの実習を通して構造化プログラミングの手法を学習している。ところが、**VB**を扱った「プログラミング」の教科書にも、市販の**VB**のテキストにもメインルーチンが記述されていない。そこで、構造化プログラミングの手法で作成する流れ図をベースにして、**VB**で作成されるプログラムの基本形を考えることにした。**VB**

で作成されるプログラムの実際の動作から、一度読みでサブルーチンによりプログラムが分割された図3のような流れ図になると考え、これを基本形とすることとした。メインルーチンは、実際にはコーディングの必要はなく、終了処理がなくてもプログラムは作成できるのだが、今回作成するテキストでは、生徒の既習事項を活かすためメインルーチンを示し、終了処理も入れることとした。

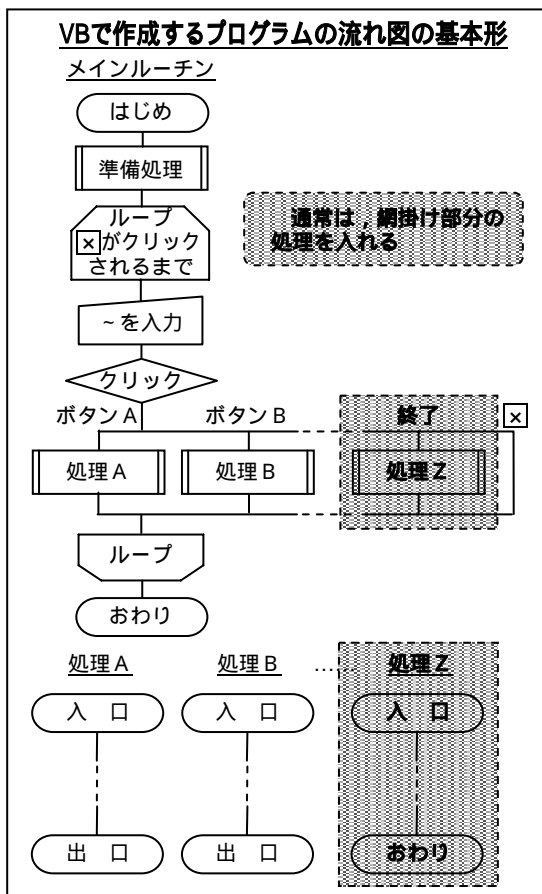


図3 VBで作成する流れ図の基本形

- (4) ネットワーク環境で実習を行うにあたっての準備
 ア .NET Framework のランタイムセキュリティポリシーの設定

本校では生徒が実習する場合、ネットワークドライブを利用してサーバーにファイルを保存している。そこで、プロジェクトファイルをサーバー上に作成しようとする時、図4のようなメッセージが表示される。これを無視し、そのままプロジェクトファイルを作成して、プログラムを作成し実行すると、異常終了してしまうことがある。これは、.NET Framework の初期値設定がネットワーク環境で動作することになっていないため、サーバーとの信頼関係が無いからである。プログラムをサーバー上に置いて、実行すると動作が保障されず、異常終了してしまうのである。したがって、サーバー上にプロジェクトファイルを作成する場合は、.NET Framework のランタイムセキュリティポリシーを変更し、サーバーを信頼させる必要がある、実習を行うにあたっては、クライアントパソコンに対して、次の作業をしておく必要がある。

.NET Framework のランタイムセキュリティポリシーの変更方法
 管理者 ID でクライアントにログオン
 [スタート] [設定] [コントロール パネル] [管理ツール]
 [Microsoft .NET Framework 1.1 Configuration]を起動
 [マイコンピュータ] [ランタイムセキュリティポリシー] [コンピュータ] [コードグループ]
 [All_code] [LocalIntranet_Zone]を右クリックし、[新規作成]を選択
 [名前]欄に任意の文字を入力し、[次へ]をクリック
 [コードグループの種類]欄を "URL" にし、[URL]欄に信頼する保存先を入力し、[次へ]をクリック
 [既存のアクセス許可セットを使用]を選択し、[Full Trust]を選択し、[次へ]をクリックし、[完了]をクリック

イ クライアントのユーザアカウントの設定

本校では、生徒用のアカウントには、管理者権限を与えていない。管理者権限のないアカウントでクライアントパソコンにログオンし、プログラムを作成し、コンパイルしようとする時、アクセスが拒否されコンパイルできない。これは、Visual Studio .NET でソフトウェアの開発を行うためには、管理者権限のあるアカウントか、debugger user のグループに属するアカウントでログオンする必要がある

からである。本校では、生徒のアカウントに管理者権限を与えていないため、生徒のアカウントのグループを debugger user のグル

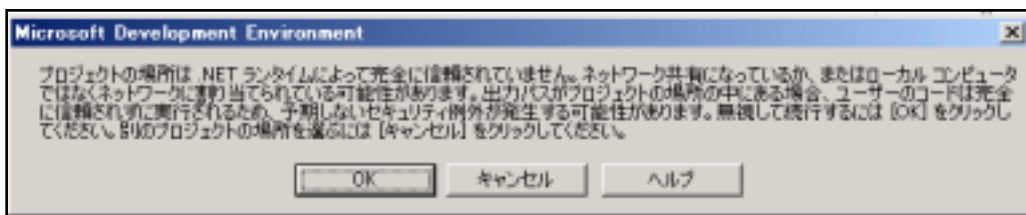


図4 ネットワークドライブへプロジェクトファイルを保存するときのメッセージ

ープ追加しておくことが必要であり、クライアントパソコンに対して、この作業をしておく必要がある。

2 テキストの構成と内容

(1) ソフトウェア開発の演習のための基本的な内容

テキストの構成と内容を考えるにあたり、ソフトウェア開発の演習のために必要な **VB** の基本的な内容について、次のように考えた。

ソフトウェア開発の演習は、それまで経営情報に関する科目で習得した知識・技能を総合的に活用し、体験的に学習するものである。したがって、新たなプログラム言語の知識・技能を習得することが主たる目的ではない。そこで、生徒がこれまでソフトウェア開発の演習で作成してきた作品(表2)が作成できるように、ファイルの処理を中心として、全商情報処理検定1, 2級プログラミング部門で扱うプログラムが作成できるような内容を、ソフトウェア開発の演習のための基本的な内容とすることとした。また、**VB.NET** では、データベースを利用したプログラムや **Web** ページ用のプログラムなど、様々な形態のプログラムの作成も可能であるが、学習時間の制約から、スタンドアロンで動作する対話型のプログラム(**Windows** アプリケーション)のみを取り上げることとした。

(2) **COBOL** プログラミング既習事項の活用

私自身、これまで最初に **FROTRAN** を学習し、次に **COBOL**、そして、今回 **VB** を学習した。**VB** を学習するにあたって、**COBOL** と対比させ特徴を

表2 過去の生徒の作品例

	販売管理システム	レンタルビデオ管理システム	テーマパーク売上管理システム
内容	日々の売上データを一定期間集計し、商品別売上一覧表、月間売上ランキングなどを出力	貸出、返却、購入などのトランザクションデータのマスターファイルの更新をすると同時に、貸出ランキング、料金収入集計表などを出力	入場券の発券処理、入場者の集計処理や売上の集計処理
技法	データチェック、ファイルの照合、整列、グループトータル	データチェック、ファイルの更新・照合、整列、グループトータル	データチェック、ファイルの照合、整列、グループトータル

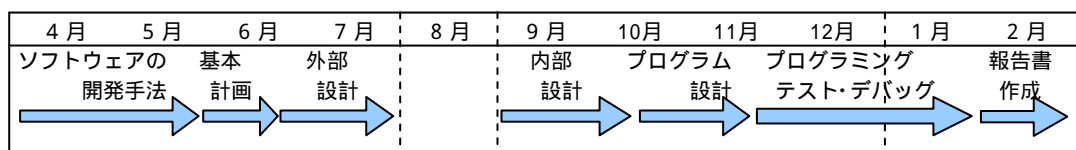


図5 ソフトウェア開発演習の年間スケジュール

整理するなどし、理解を深めていった。このように、一つの手続き型のプログラム言語のプログラミング技術を習得していれば、他のプログラム言語を学ぶ際には、既習事項が活かされ、比較的容易に理解することができる。また、流れ図を作成する技術は基本的には共通の技術であり、プログラム言語に関係なく活かすことができる。**VB.NET** は、オブジェクト指向を意識した言語ではあるが、既習の構造化プログラミングの考え方が活かすことができ、流れ図は同じように作成可能である。

(3) テキストの作成にあたっての工夫

ア ソフトウェア開発の演習で用いるための工夫

ソフトウェア開発の年間スケジュールは、図5のようになっており、この中で **VB.NET** の学習に多くの時間を割くことは不可能である。したがって、10時間程度(3単位で1か月以内)で学習を終えられるようにする必要がある。そのため、必要最小限度の基本的な内容に絞り、量的に多くならないように注意しなければならないが、基本的な内容に基づいて、ファイル処理、印刷処理、エラー処理については、必須と考え、また、**Excel** の利用についても取り上げることとした。**Excel** を利用することで、テーブル(配列)、索引ファイルやデータベースと同様の処理が実現でき、また、帳票の印刷も容易に行えるためである。これにより、これまで生徒の希望として多くあった、索引ファイルやデータベースの使用に対応することができる。

イ **COBOL** プログラミングの学習をした本校生徒が用いるための工夫

作成したプログラムが実行されるまでの流れや、プログラミングの手順(図1)、命令文など、**COBOL** と **VB** を対比させ既習事項が活きる記述をし、例示するプログラムも **COBOL** で実習した課題(売上一覧表作成、図書マスタファイルの更新)とほぼ同様のものを使用する。このことにより、課題内容の理解

が容易にでき、流れ図も容易に読み取ることができ、効率的に学習を進められるようになると考えた。

COBOL と対比した記述例(テキスト本文 3-4 コーディング)

それでは、流れ図にしたがって、更新処理のサブルーチンのコードを書いていきますが、その前に変数名(COBOL ではデータ名)の定義を行います。図33のPrivate Sub Button1_Click(ByVal...の前に、このプログラムで使用する変数を流れ図で確認して、次のように定義してください。

- Dim 変数名1[変数名2,...] As データ型の種類
- ・変数名は、英数字でつけます。
- ・変数名に“-“は使えません。
- ・変数名は複数個まとめて定義できます。
- ・データ型の種類は、String(文字型)、Integer(整数型)、Single(単精度浮動小数点数型)、Double(倍精度浮動小数点型)などがあります。(String は COBOL の英数字項目や日本語項目に対応します。COBOL の数字項目に対応するデータの型もありますが、VB では数値を扱う場合は、整数と小数で分けて定義して使います。数字編集項目に対応する型はありません。出力のときに編集して出力します。)

更新処理のサブルーチンは、図34のようになります。ファイルを扱う場合は、COBOL と同様に最初にファイルを開いておき、最後にファイルを閉じる必要があります。

図6 テキスト本文の記述例

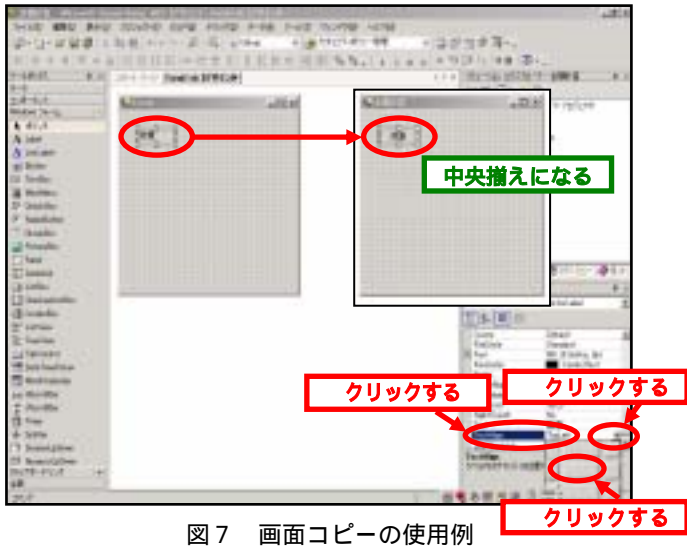


図7 画面コピーの使用例

ウ 自学自習できるための工夫

本校での実習環境(ネットワークドライブなど)に合わせた記述を行い(図6)、視覚表現を多用する(図7)。以上の点を考慮し、表3に示すような構成と内容でテキストを作成した。

(4) テキストの試用、改善

ア テキストの試用

平成16年12月中旬の6日間、全商情報処理検定1級プログラミング部門の学習を終えた情報処理科3年生と情報システム科2年生から参加希望者を募り、作成したテキストを試用するためVBの学習会を放課後に行った。学習会では、テキストの改善のため、次のような学習記録用紙に毎回記入させ、最終日には全体を通してのアンケートを実施した。

学 習 記 録 用 紙	
月 日()	名前 _____
1. 学習範囲	_____ ページ ~ _____ ページ
2. 学習時間	_____ 時間 _____ 分
3. 今日の学習内容は理解できましたか。	ア.よく理解できた イ.だいたい理解できた ウ.あまり理解できなかった エ.全く理解できなかった
4.	よくわからなかった用語、内容(ページ数と具体的に書く)
5.	わかりやすかったこと(ページ数と内容を書く)
6.	追加して欲しいことがありましたか(説明、命令、文法など)
7.	感想(感じたことを何でもいいですから書いてください)

表3 テキストの構成と内容(改善前)

構 成	内 容
第1章 Visual Basic.NET とは 1-1 .NET Framework と VB.NET 1-2 VB でのプログラム作成手順	・VB.NET などで作成され、.NET Framework 上で実行されるプログラムと COBOL などのコンパイラ言語の実行までの流れを対比して整理 ・COBOL と VB のプログラミング手順について対比させ整理 ・流れ図の基本形の提示
第2章 VB.NET でプログラムを作る 2-1 これから作るプログラム 2-2 フォームの作成 2-3 コーディング 2-4 テストラン・デバッグ	・1学年で最初に行う COBOL 実習課題(売上一覧表の作成)と同様の単純な画面入出力のプログラム作成 ・プロジェクトファイルの作成からテストランまでの Visual Studio.NET の操作説明
第3章 ファイル処理(順次編成ファイル) 3-1 ファイルの更新処理の流れ図 3-2 入力ファイル(テキストファイル)の作成 3-3 フォームの作成 3-4 コーディング 3-5 テストラン・デバッグ	・2学年で行う COBOL 実習課題(図書マスタファイルの更新)のプログラム作成 ・テキストファイル(CSV ファイル)の作成 ・テキストファイルの OPEN から入出力の処理
第4章 Excel の利用 4-1 Excel の参照設定 4-2 Excel の利用と Excel ファイルの出力 4-3 Excel ファイルの入力と上書き保存	・Excel を利用するためのプロジェクトの参照設定 ・新規に Excel ブック、シートを作成し、関数を利用した処理を行い、Excel ファイルに名前を付けての保存するプログラムの作成 ・Excel ファイルからデータを入力し、上書き保存するプログラムの作成
第5章 印刷処理 5-1 COBOL と同等の印刷 5-2 Excel を利用した印刷	・1学年で最初に行う COBOL 実習課題(売上一覧表の作成)のプログラム作成 ・文字列の編集 ・Excel を利用した罫線のある表形式(帳票)の印刷するプログラムの作成
第6章 エラー処理 6-1 入力データのチェック 6-2 例外処理など	・入力データのチェックを行うプログラムの作成 ・実行時にエラーが起こった場合の処理を行うプログラムの作成 ・ファイルの照合

イ アンケート結果、学習記録用紙の分析

テキストを作成するに当たって工夫した点について確認して見た。

アンケートの結果は次のとおりであった。

テキスト試用後のアンケート結果（回答5人）

- 学習時間合計 平均 404 分 (50×8 時間相当)
- VB でプログラムを作成できるようになったと思うか
思う 4 思わない 1
- COBOL の学習が活かされる内容になっていると思うか
思う 5 思わない 0
- 文字を小さくし、ページ数が少ない方がよいか
小さい方がよい 2 大きい方がよい 3
- 行間を狭くし、ページ数が少ない方がよいか
狭い方がよい 3 広い方がよい 2
- 読みやすかったか
読みやすかった 3 読みにくかった 2
- 読みにくかった、わかりにくかったところ
・文字の区別 I と 1 と 1 など
・ページ数は少ない方がいいが、文字を大きくしてほしい
・文字にも色をつけて欲しい
- 次の内容は必要か

複数フォームの使用	必要	5	不要	0
ファイルの指定	必要	5	不要	0
日付処理	必要	4	不要	1
時間処理	必要	4	不要	1
リストの使用	必要	5	不要	0

学習時間は、平均で約 400 分(50 分×8 回相当)となっており、想定した時間内であった。アンケートの結果や学習記録用紙の次のような記述から、COBOL プログラミングの既習事項が活用されていることが確認できた。

生徒の学習記録用紙から

- ・第1章の内容で COBOL と VB の違いが簡単に書いてあり、分かりやすかった。
- ・初めて VB に触れてみたけど、COBOL に似ていて、親しみやすかったです。
- ・18 ページの b=a というのが、MOVE 文と比較してあって分かりやすかったです。

また、生徒が各自でテキストを読み進めながら、実習に取り組む様子や、生徒の質問は操作に関するものが多く、内容に関するものはほとんど無かったことから、自学自習できることが確認できた。

ウ 生徒の視点からの改善

生徒の学習記録用紙に、Form の Text プロパティの変更、Label の Text Aligned プロパティの変更など、説明が分かりにくいと指摘があった。そこで、本文を見直すことはもちろん、画面のコピーを追加したり、コピーの中に説明を加えたりした。

また、テキスト試用後のアンケートから、フォントサイズを少し大きくし、行間を少し狭くした。また、英数字を見分けるのが難しいという声があった

ので、原則として英数字は、明朝体からゴシック体に変更した。

生徒の学習記録用紙から

- ・P.7 の図 12 の「ここをクリック」という部分の出し方の説明が Text Aligned をクリックしても図のように表示されない。図のように出そうとしたら Text Aligned の横の を押さないといけないのでそれを足してほしい。
- ・P.8 Form の Text プロパティの変更がはじめどこかわからなかった。

エ 内容の追加

テキスト作成後、追加すべきではないかと考えた内容を五つ(複数フォームの使用、入出力ファイルの指定、日付処理、時間処理、リストの使用)アンケートで生徒に尋ねたところ、五つとも追加すべきと答えた。そこで、再度これらの内容を加えるべきか検討をした。

入出力ファイルの指定は、普段からソフトウェアを利用する際には、ファイルを開いたり、保存したりするときに必ず行っている。そこで、「第3章 ファイル処理」の中に追加することとした。複数フォームの使用も、ソフトウェアを利用すると通常複数のウィンドウが開くようになっているので、「第4章 Excel の利用」の中で例示している図書マスタファイルの更新を行うプログラムに追加することとした。日付処理、時間処理は、ビジネス用のソフトウェアでは、日付や時間のデータは重要であり、様々なデータに付加され処理されているので、「第5章 印刷処理」の中で例示している売上一覧表を作成するプログラムに日付処理を追加することとした。時間処理については、日付処理とほぼ同様に処理できるので、追加を見送った。リスト処理については、都道府県の入力や日付の入力など実際によく使われている。しかし、これを使わなくてもプログラムの作成は可能であり、また、今回テキストの中で例示したプログラムでは、必要性が低いと思われるので追加を見送ることとした。

以上のように、内容を追加し、最終的なテキストの構成と内容を、表4のように改善した。

成果と課題

1 成果

本研究の成果として、次のことがあげられる。

表4 テキストの構成と内容(改善後)

構 成	内 容
第1章 Visual Basic.NETとは 1-1 .NET FrameworkとVB.NET 1-2 VBでのプログラム作成手順	<ul style="list-style-type: none"> ・VB.NETなどで作成される.NET Framework上で実行されるプログラムとCOBOLなどのコンパイラ言語の実行までの流れを対比して整理 ・COBOLとVBのプログラミング手順について対比させ整理 ・流れ図の基本形の提示
第2章 VB.NETでプログラムを作る 2-1 これから作るプログラム 2-2 フォームの作成 2-3 コーディング 2-4 テストラン・デバッグ	<ul style="list-style-type: none"> ・1学年で最初に行うCOBOL実習課題(売上一覧表の作成)と同様の単純な画面入出力のプログラム作成 ・プロジェクトファイルの作成からテストランまでのVisual Studio.NETの操作説明
第3章 ファイル処理(順次編成ファイル) 3-1 ファイルの更新処理の流れ図 3-2 入力ファイル(テキストファイル)の作成 3-3 フォームの作成 3-4 コーディング 3-5 テストラン・デバッグ	<ul style="list-style-type: none"> ・2学年で行うCOBOL実習課題(図書マスタファイルの更新)のプログラム作成 ・テキストファイル(CSVファイル)の作成 ・テキストファイルのOPENから入出力の処理 ・入出力ファイルの指定(追加)
第4章 Excelの利用 4-1 Excelの参照設定 4-2 Excelの利用とExcelファイルの出力 4-3 Excelファイルの入力と上書き保存	<ul style="list-style-type: none"> ・Excelを利用するためのプロジェクトの参照設定 ・新規にExcelブック、シートを作成し、関数を利用した処理を行い、Excelファイルに名前を付けての保存するプログラムの作成 ・Excelファイルからデータを入力し、上書き保存するプログラムの作成 ・複数フォームの利用(追加)
第5章 印刷処理 5-1 COBOLと同等の印刷 5-2 Excelを利用した印刷	<ul style="list-style-type: none"> ・1学年で最初に行うCOBOL実習課題(売上一覧表の作成)のプログラム作成 ・文字列の編集と日付処理(追加) ・Excelを利用した罫線のある表形式(帳票)の印刷するプログラムの作成
第6章 エラー処理 6-1 入力データのチェック 6-2 例外処理など	<ul style="list-style-type: none"> ・入力データのチェックを行うプログラムの作成 ・実行時にエラーが起こった場合の処理を行うプログラムの作成 ・ファイルの照合

- ソフトウェア開発の演習で活用できるテキストを作成することができた。作成にあたって考慮した事項をほぼ反映した内容とすることができた。
 - ・テキストの学習時間は、内容の追加を行ったが、10時間程度となるのではないかと考える。
 - ・作成したテキストが既習のCOBOLプログラミングの知識・技能を活かせることを確認できた。
 - ・作成したテキストで自学自習できることを確認することができた。
- VB.NETをネットワーク環境で利用する場合、クライアントの設定が必要であることが分かり、その方法について明確にすることができた。

2 課題

課題として、次のことがあげられる。

- 少数の生徒による試用であったため、実際に活用する際には、今回確認できなかった課題が発生する可能性があり、継続して改善を図る必要がある。
- テキストの内容として追加すべきと思われるリストの利用を取り上げられなかった。例示プログラムの再検討を行い追加したい。今回は外部分類については触れなかったが、大量のデータを扱う場合は必要であり、今後検討をする必要がある。

- このテキストでは、取り上げなかったデータベースやWebページの利用などの多くの内容の取り扱いについて検討が必要である。生徒の開発したいソフトウェアの変化や意欲を喚起する内容に対応するためのテキストの改善が必要である。データベースやWebページの利用などの発展的な内容については、別冊を用意して指導するか、市販のテキストを利用するかなど、ソフトウェア開発の演習に有効なテキストの選択の検討が必要である。

参考文献

- ① ITフロンティア『Visual Basic .NET 逆引き大全 500の極意』 秀和システム 2003
- ② (株)オフィスエム『図解標準最新 Visual Basic .NET ハンドブック』 秀和システム 2002
- ③ 林 晴比古『新 Visual Basic .NET 入門ビギナー編』 ソフトバンクパブリッシング 2003
- ④ 矢嶋 聡『VB .NET プログラミングリファレンス』 リックテレコム 2001
- ⑤ 吉野弘一『商業科教育法—21世紀のビジネス教育—』 実教出版 2002