

# オブジェクト指向による情報システム設計演習における 開発支援環境

*A Study of Development Support System  
for Object-Oriented Design Exercise of Information Systems*

槻 木 公 一

## 要約

本学の情報システム学科では、オブジェクト指向の概念と設計から開発までの展開を具体的に理解するために、情報システム設計開発の演習科目を設けている。この演習の後半部分では OOP によるプログラム作成を行っているが、学生と教員双方に様々な負荷が増大してきた。

ここではこの問題を解決すべく、短時間にプログラムの初学者でも簡単な情報システムを OOP で構築できる開発支援環境について述べる。本開発支援環境は、予め作成したスーパクラスを提供し、RDB を不要としたオブジェクトの永続化、継承による実装プログラム作成量の削減、分かりやすいデバッグ支援を実現する。

まだ半期 1 回の演習に供したところであり、定性的な評価にとどまるが、OOP による実装負荷を具体的に軽減することができたと評価する。今後、種々の作成事例を増やしてさらなる改修を行い、有効性の定量的評価を進める必要がある。

キーワード： オブジェクト指向 OOP 開発演習 支援環境

## 1. はじめに

情報システム設計開発の演習科目において、オブジェクト指向の概念と設計から開発までの展開を具体的に理解するために、演習の後半部分では OOP によるプログラム作成を実施してきたが、主として演習時間の不足に起因する様々な問題が生じてきた。

本研究では、従来のプログラム作成の方法を見直して、プログラムの初学者でも短時間に簡単な情報システムを OOP で構築できる開発支援環境を提案する。

## 2. オブジェクト指向による情報システム設計演習の目的

オブジェクト指向設計 (Object-Oriented Design : 以下 OOD と呼ぶ) を学習し、その成果として個々に簡単な情報システムの設計演習を実施する。続けて、オブジェクト指向プログラミング (Object-Oriented Programming : 以下 OOP と呼ぶ) による開発演習を通して、設計した情報システムを実装して評価する。設計から実装までを通して全体の流れを学習する PBL (Project-Based Learning) 方式の演習であり、オブジェクト指向の概念に基づいた情報システム構築を具体的に理解することを目的とする。

---

\* TSUKIGI, Kouichi [情報システム学科]

### 3. 現状の問題点

#### (1) OOP の学習時間の不足

本演習は専門科目の演習として、3年次前期に開講している。2年次後期までに設定されたプログラム関係の科目として、初級、中級レベルのC言語と初級レベルのVisualBasic (VB Express2005) 言語の演習科目がある。C言語はOOPではなく、VBの科目はプログラミング初学者向けであり、授業回数も少なく自力でプログラム作成できる学習内容ではない。さらに、いずれも必修科目ではないので、本演習に必要なOOPに関する知識はこの演習の中で学習させざるを得ない。

#### (2) RDBの利用とOOPの一貫性の限界

情報システムとして実現するためには、何らかの形で「オブジェクトの永続化」への対応が不可欠であり、一般的にはオブジェクト指向データベース (Object-Oriented DataBase) の導入が必要となる。しかし、OODBを扱うOODBMSは商用として開発されたものが多く、その導入の敷居は高い。一方、通常よく用いられているRDBMSは、商用からフリーのものまでその価格、性能、DB規模において選択肢が多く、現実のオブジェクト指向の情報システム開発にも利用されている。本演習においても従来からACCESSを用いているが、OOPの観点からは一貫性を保つことができない。

加えて、RDBMSを利用するためには、オブジェクトの属性値とRDBのタプルとの相互変換とRDBの「主キー」の設計が必要となる。また、オブジェクトの検索のために、SQL文によるタプル検索処理を組み込むことになる。

#### (3) OOD理解のための実装内容の不足

限られた演習時間内で設計する情報システムでは、設計対象のクラス数を少なくした比較的単純な内容にならざるを得ない。その結果、オブジェクト指向の重要な概念である「継承」に関しては机上での学習に留まり、OOPによる実装を通して理解を促進するということできていない。

### 4. 演習の実施面における課題

#### (1) 学生の学習負荷の増大

限られた演習時間内 (2コマ、15回) に、OOD、OOP、RDBの設計 (特に主キーの概念) とSQL処理の学習が必要となり、演習の後半部分に割り当てられたOOPによる開発実装に充当する時間数が不足している。さらに、プログラム作成においては、OOPの学習不足と重なってコンパイルエラーおよび実行時エラーが頻発している。さらに、プログラム作成の経験が少ないため、発生したエラーの原因を特定することが困難であり、プログラム完成までに時間がかかってしまう。その結果、本演習の成果物であるべき個々に設計した情報システムの完成度が低下し、一部の学生では学習意欲が低下する状況が生じている。

#### (2) 教員側の指導負荷の増大

開発実装のための時間不足のために、個々の学生への作成指導時間が長くなり、指導教員数を増員せざるを得ず、また、指導側においてもエラー発生の原因特定に時間がかかり、場合によっては演習時間外での対応をせざるを得ない状況にあった。

個別に設計した情報システムの実装時の手戻りとエラー発生をできるだけ防止するため、実装に着手する前にすべてのクラス構造と機能、さらにRDBのテーブル設計と主キーの妥当性をチェックする必要もあり、これも演習時間外に教員が対応せざるを得なくなっている。

## 5. 演習環境の改善と開発支援環境の構築方針

専門科目の演習には履修内容が異なる複数の演習があるので、他の科目への影響を及ぼすことなくこの演習科目内だけで改善する必要がある、下記の改善目標を設定した。

「演習の前半部分で、OOD の基本的な知識を取得済であることを前提とした上で、最小限の OOP 知識だけで、プログラム作成の経験がほとんどない学生が、学習意欲を損なわないように短期間で、かつエラー発生も少なく自分が設計したシステムを開発できること」

そのために本演習の後半部分を対象とした開発支援環境を VB で作成することとし、下記の構築方針を定めた。

- ・個別に設計する情報システムの形態やクラス数、構造を制限してパターン化し、類似事例に基づいて OOD/OOP の理解を促進する。
- ・OOD における「継承」を理解するための枠組みを新たに組み込む。
- ・OOP 言語の部分的な理解でもプログラムが作成できるようにし、実質的なプログラム作成量をできるだけ少なくする。
- ・OOP と直接関係の薄い RDB 知識を不要とし、別途「オブジェクト永続化」の簡易な実装方法を組み込む。
- ・プログラム作成と実行における各種のエラー発生を最大限に防止するとともに、エラーが発生した場合は、その原因を容易に特定できるようにする。

## 6. 具体的な改善策

- (1) 設計対象として、新規オブジェクトの生成、オブジェクトの条件検索、該当オブジェクトの属性の読み出しと変更処理をすべて含むシステムとする。

例：顧客管理をした各種予約システム、販売システム、貸出システムなど。

- (2) 個別設計システムのベースとしてスーパークラス群を提供し、これらを継承するサブクラス群のみを個別作成対象として、新規に作成するコード量を削減するとともに、「継承」の概念を OOP で具体的に実装する。

- (3) 開発支援環境では、オブジェクト永続化処理と SQL 処理に該当する検索処理などの基本部分を事前に作成してスーパークラス群に組み込み、サブクラスのコード作成量を削減する。具体的には、クラス毎に CSV 形式のテキストファイルを割り当て、属性名およびオブジェクトの属性値をすべて文字列データとして扱うことによって、擬似的にオブジェクトの永続化を実現する。検索処理では、SQL の Where 句の条件部分を、単一の条件句に相応する機能（メソッド）の組み合わせに置き換えてスーパークラスから提供し、SQL を利用しない検索機能を実現する。

- (4) OOP 言語は、初学者でも比較的理解しやすい VB とする。しかし、コンパイル時や実行時にエラーが発生すると、その原因の判別が困難である場合が多い。エラーが多発し修正に時間がかかると、プログラム作成意欲が失われるのはもちろんのこと、作成指導の負荷が増大する。そのため、上記のスーパークラスの提供機能（メソッド）の中でできる限りユーザレベルのエラーチェックをコンパイル時と実行時に行う。

- ・個別に作成するクラスに含まれるデータの属性名は、対応する CSV ファイルの第一行目を項目名欄とし、このデータから属性名を自動設定する。
- ・半角文字と全角文字の混在に起因するエラーを防ぐため、半角文字に変換できる全角文字、英数字や記号文字はすべて事前に半角文字に変換して処理を行う。

- ・ SQL の代替機能として単一の条件句に相応する機能をスーパークラスに設けるが、比較演算子に対応する文字列は、全角文字であってもまた記号の順序が入れ替わっていても、正しく処理を行うようにする。
  - ・ 空白文字の有無を考慮せずに検索条件を設定し、条件不一致となるエラーが多いことから、データとして半角もしくは全角空白文字を含む文字列の処理において、事前にすべて空白文字を削除する。
- (5) エラーの原因を判別しやすくするため、できるだけ事前チェックを行って、実行時エラーの発生を避ける。例えば、指定されたファイル名や属性名の存在チェック、条件設定エラーなど発生を把握したエラーはすべてテキストファイルに保存しておく。VB の実行終了後にこのファイルを読み出せるようにして、エラー原因の追究を容易にした。このテキストファイルは追記形とし、作成期間中は継続してエラー記録を取得できるようにした。
- (6) SQL の知識がなくても簡単に目的のオブジェクトを検索するため、単体の条件句に相応する機能（メソッド）をスーパークラスに設定するとともに、検索結果の処理においても、コレクション処理（配列やセットの処理）ではなく、順番にオブジェクトを取り出す単純な逐次処理で済むような機能を用意した。コレクション処理における制御構造を不要にして、VB の初学者でも容易に開発できるように工夫した。
- (7) その他の作成上の留意事項として、OS や開発言語などのバージョンアップへの対応を考慮しておく必要がある。バージョンが変わるたびに指導内容やテキスト修正が必要となり、市販の教材が使えない場合はその修正負担が大きい。また、経費面からバージョンアップへの対応は一般的に遅れ気味であり、更新間隔が長くなると段階的なバージョンアップができなくなる。
- 本開発支援環境の作成においても、開発言語のバージョンアップの度にプログラムを改修する必要がないように、できるだけ配慮しておく必要がある。VB から ACCESS を利用する場合には、参照モジュールとのバージョンの整合が常に必要であったが、テキストファイルを使用する今回の開発支援環境では、すべて VB 言語だけで作成することができ、VB2005、VB2008、VB2010、VB2012 へとバージョンアップしても、自動変換で使い続けられることを確認した。（現在は、まだ VB2005 で使用中である。）

## 7. 開発支援環境の概要

開発支援環境としては、2つのスーパークラス、TargetManager クラス（以下、TargetManager と略す）と Target クラス（以下、Target と略す）を組み込んだ VB の開発環境を提供する。この中には、さらに CSV ファイルの設定フォルダを明示するためのひな形ファイルと、サンプルコーディングを示したテキストファイルも含む。この環境を利用した OOP 開発では、提供されたスーパークラスを継承するサブクラスとして、個別に設計した実装クラスを作成することになる。

Target は、オブジェクトの永続化を必要とする実装クラスのスーパークラスであり、TargetManager は、Target のオブジェクトを一括管理するスーパークラスである。このクラスを継承したユーザ独自のオブジェクト管理クラスを、全体でひとつだけ作成し、永続化を必要とするそれぞれの実装クラスへは、このオブジェクト管理クラスの各オブジェクトで対応する。

図 7.1 に予約システムを事例としたクラス図を示す。この事例において、プログラムを作成するクラスは、TargetManager と Target を除いた 6 つの実装クラス（サブクラス）となる。

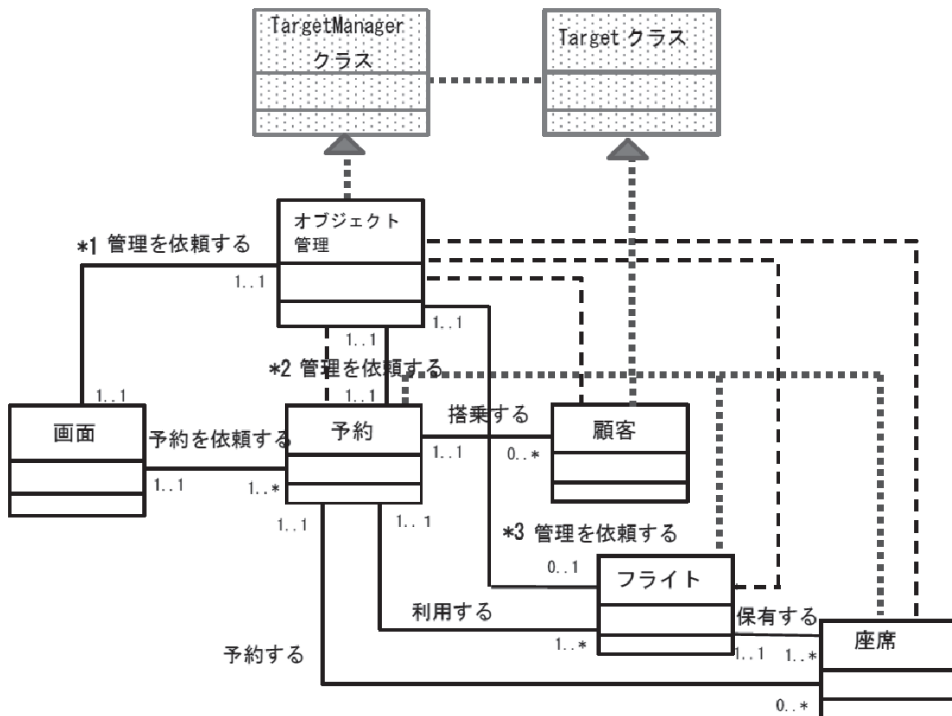


図7.1 フライト便の座席予約システムを事例としたクラス図

## 8. オブジェクト永続化の考え方とデータファイルの構造

CSV 形式のテキストファイルを用いてオブジェクトを擬似的に永続化するため、個別オブジェクトとテキストファイルの各行データとの対応関係を、設計時に学生に明確に示しておく必要がある。

- (1) 顧客、会員、商品など、明確に他と区別できる対象物であり、利用する単位も対象物そのものである場合は、現実の対象物そのものをひとつの個別オブジェクトとして捉える。
- (2) 現実の対象物はひとつであるが、利用する単位が日別あるいは時間別に分かれる場合は、利用する単位をひとつの個別オブジェクトとして捉える。いわば「分身オブジェクト」として考える。例えば、列車や航空の座席（指定席）、ホテルや旅館の部屋、レンタカーなどがある。
- (3) 同種の対象物が複数あって、利用する単位が同種のものの中のいずれでもよい場合、例えば列車や映画館の自由席とか、図書とか DVD などの貸出などの場合は、下記の 2 つの捉え方がある。
  - ・同種の対象物であっても、(1)と同様、それぞれを個別のオブジェクトとして捉える。
  - ・同種の対象物をまとめて一つのオブジェクト、いわば「集合体オブジェクト」として捉え、同種の対象物の存在をこの「集合体オブジェクト」の中に個数の属性を設定して表現する。

## 9. 制約事項

CSV ファイルはメモ帳や EXCEL で簡単に作成できるが、RDB ではないのでレコードの重複やデータ項目名の重複がないように注意して作成しなければならない。また、型の宣言がなく、す



べて文字列として扱うので、他の型のデータとして処理する際には明示的に型変換が必要になる場合がある。ただし、TargetManager が提供する検索条件の大小比較演算では、文字列からの数値変換や日付変換を自動的に行う。

複合した検索条件を設定するために、まず条件の AND 論理グループ、OR 論理グループを別個に設定し、さらにこれらグループ間を AND または OR 論理で結合するレベルまでを実現した。過去の本演習における SQL を使用したすべての作成事例には対応できることは確認したが、Where 句で記述できるような柔軟な検索レベルは実現できていない。

また、前述のようにテキストファイルを用いた擬似的なオブジェクト永続化であるので、複雑なオブジェクト構造には対応できない、目的のオブジェクトを見つけるとしても結果的には RDB のような文字列の「キー」に頼らざるを得ないという制約もあるが、この点においても過去の演習での作成事例への適用には問題がないことを確認した。

Target を継承する実装クラスにおいては、個別設計された検索処理や更新処理を作成するので理解しやすいが、TargetManager の実装クラスの中には個別設計の内容とは無関係の機能を作成しなければならない。検索条件に合致する Target のオブジェクトそのものの取得機能は TargetManager の機能にあるが、Target の実装クラス（サブクラス）のオブジェクトとして取得するには逆キャストとなるので、スーパークラスの機能の中には実現できない。そのため、TargetManager の実装クラスの中にこの取得機能を作成し、検索条件に合致した Target の実装クラスのオブジェクトを擬似的に取り出さなければならない。この機能は Target の実装クラス別に必要となるが、クラス名だけが異なるコードであるので、サンプルコードを示して作成させている。

## 10. 作成事例

演習の後半部でまず、OOP と開発支援環境の理解を兼ねて、TargetManager の実装クラスを 1 つ、Target の実装クラスが「顧客クラス」1 つのだけの「住所録システム」を全員に作成させてから、個別の設計開発を進める。個別設計では、「住所録システム」に、さらに Target の実装クラスを少なくとも 2 つ以上追加し、オブジェクトの読み出し、修正、挿入などの処理を行う内容にすることを要求している。

本開発支援環境での作成事例はまだ少ないが、「高速バス予約システム」「レンタカー予約システム」「総合病院予約システム」「楽器販売システム」「居酒屋予約システム」「Web マネー購入システム」などがある。

## 11. 評価

演習 15 回分の各回の割り当てを従来の内容と比較すると、OOP の理解と支援環境説明に 1 回分増やし、逆に OOP の作成回数は 1 回分減らしたが、個別に設計した内容を全員が作成することができた。学生が作成した実質的ステップ数は 100 ステップ前後であり、従来に比べて 20% 程度の削減に止まっている。これは SQL の Where 句では一行で記述できた条件句を、今回は個々に分割して記述していることによるものと考えられる。しかし、理解しやすいプログラムコードとなったことで、発生するエラーの数が激減し、演習時間中での作成プログラムの完成度も上がり、時間外の指導も皆無となった。この開発支援環境において継続的に取得しているエラー記録を見ると、項目名指定や条件句設定において、文字列に関わる単純なエラーが重複して発生する

ことも少なくなっている。これは、開発支援環境が独自に出力するエラー表示に、その内容が詳細に説明されていることによって、エラー箇所の発見がかなり容易になったと推測される。

一方、この開発支援環境では補足できない VB 環境でのエラーも当然発生している。その種類とか発生数はまだ定量的には把握しておらず、本開発支援環境の有効性の評価に関しては、さらなる作成事例を積み上げる必要がある。

## 12. あとがき

情報システムの設計開発演習において、従来の OOP によるプログラム作成時の問題を見直し、プログラムの初学者でも対応できる開発支援環境を作成して実際の演習に供した。まだ半期 1 回の適用であり定性的であるが、本開発支援環境を利用することにより、OOP による実装負荷を具体的に軽減することができたと評価する。今後、種々の作成事例を増やしてさらなる改修を行い、有効性の定量的評価を行っていく。

