2008－CE－96 (7)
2008／10／25

オブジェクト指向プログラミング教育における採点支援システムの開発とその評価

高野 辰之† 宮川 治‡ 小濱 隆司‡

† 東京電機大学大学院 情報環境学研究科
‡ 東京電機大学 情報環境学部

概要: 情報環境学部ではオブジェクト指向プログラミングの教育が行われている．そこでは講義と演習を行い，試験では学生にクラス図とＡＰＩ仕様からスクラッチでプログラムを作成させ，提出させている．そこで，プログラムの採点業務を支援するため，業務分析とシステム開発を行った．このようなシステムの開発事例は少ない．本稿では，学生が提出したプログラムの採点を支援する手法，および開発したシステムによる採点業務の効率化について報告する．

Development and Evaluation of a Scoring Support System for Object-Oriented Programming Education

Tatsuyuki TAKANO†, Osamu MIYAKAWA‡ and Takashi KOHAMA‡

†Graduate School of Information Environment, Tokyo Denki University
‡School of Information Environment, Tokyo Denki University

Abstract: The education of object-oriented programming is conducted in the School of Information Environment. It consists of the lecture and practice. In each examination, the students are required to make programs from scratch in reference to the class diagrams and API specifications, and submit them. Therefore, we analyzed the scoring process and developed a system to support the marking duties. Few systems have been developed with this aim. In this paper, we report on the marking support method and the efficiency of our system.

1. はじめに

本学の情報環境学部ではオブジェクト指向プログラミングの講義を実施している．オブジェクト指向プログラミングの基礎を学ぶコンピュータプログラミングＢでは平成20年度には200名近くの学生が受講した．そして，学生の評価は主にレポートとテストによって行われ，テストは学生にプログラムの仕様書を配り，白紙の状態から記述するスクラッチによるプログラミングにより作成・提出させる．

プログラミング教育において，教員の負担を軽減させるためプログラムの評価を行う技術を利用した支援システムの研究[1)2)3)4)]が取り組まれている．しかし，オブジェクト指向プログラミングや厳密なプログラムの仕様を与えてのテスト形式，スクラッチによるプログラミングなどの要素が複合している場合には従来の研究を適応することは可能だが部分的である．そのため，本研究ではこれらの要素を考

慮し，採点支援システムの開発に取り組み，その試作を行った．本稿では提出されたプログラムの採点を支援する手法と試作したシステムによる採点業務の効率化について述べる．

2. 講義について

本研究ではオブジェクト指向プログラミング教育の基礎科目にあたるコンピュータプログラミングBを開発する採点支援システムの対象とした．対象とした講義の位置づけと内容，テストの形式について説明する．

2.1 オブジェクト指向プログラミング教育の科目

本学部でのオブジェクト指向プログラミングの教育は以下の科目で構成され，プログラミング言語にはJavaを使用している．

- コンピュータプログラミングA
- コンピュータプログラミングB
- オブジェクト指向設計

コンピュータプログラミングA・Bはプログラミング教育の基礎科目であり，複数の教員が各教室にて講義と演習を実施し，各教員はWeb上に作成された同じテキストを用いている．

そして，コンピュータプログラミングBからオブジェクト指向プログラミングの題材が入った内容となっている．最後のオブジェクト指向設計は発展科目としてあり，デザインパターンも含めた内容となっている．

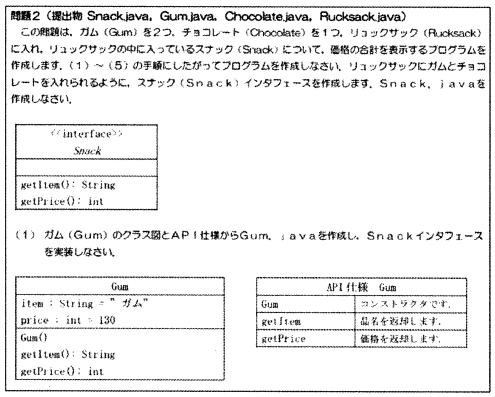
2.2 講義の内容

コンピュータプログラミングBは講義中に演習が含まれており，学生は各自ノートPCを持参して講義を受ける．受講者が約200名いるため複数の教員によってクラス分割されている．そして，各教室にスチューデントアシスタント(SA)やティーチングアシスタント(TA)が配置され，演習の際にサポートを行っている．

テキストはWeb上に作成され，その内容はオブジェクトの理解から始まり，可変長配列を利用した集約やインターフェイス，継承といった概念を学ぶ構成となっている．そのなかでUMLでのクラス図やオブジェクト図，シーケンス図などの表記を使用し，学生は特にクラス図とAPI仕様を読み目的のクラスのプログラムを完成させる演習を行う．

2.3 テスト形式

テストは中間・期末に各教員とも同一のものを実施している．学生にはクラス図やAPI仕様が記されたプリントを配布し，時間制限内にスクラッチによる目的のプログラムを作成させ，USBメモリに入れて提出させる．また，学生には不正防止の為，ネットワークへの接続は禁止している．図1は中間テストの例である．


問題2（提出物 Snack.java，Gum.java，Chocolate.java，Rucksack.java）
この問題は，ガム（Gum）を2つ，チョコレート（Chocolate）を1つ，リュックサック（Rucksack）に入れ，リュックサックの中に入っているスナック（Snack）について，価格の合計を表示するプログラムを作成します．（1）～（5）の手順にしたがってプログラムを作成しなさい．リュックサックにガムとチョコレートを入れられるように，スナック（Snack）インタフェースを作成します．Snack，javaを作成しなさい．

<<interface>>
Snack
getItem(): String
getPrice(): int

（1） ガム（Gum）のクラス図とAPI仕様からGum．javaを作成し，Snackインタフェースを実装しなさい．

Gum
item : String = "ガム"
price : int = 130
Gum()
getItem(): String
getPrice(): int

API仕様 Gum	
Gum	コンストラクタです．
getItem	品名を返却します．
getPrice	価格を返却します．

図 1: テストの例

3. システムの概要

採点支援システムは従来の採点プロセスを業務分析し，そのプロセスと採点の際にチェックする項目を明確にして，それを基にJavaで開発を行った．今回は大量にある学生のファイルを機械的に判定し，教員が目視で判断していた部分を減らすことを目的にシステムを試作した．

3.1 従来の採点プロセス

採点に関する流れとチェックする項目は以下のようになる．

1. ソースコードのコンパイル
2. 仕様の満足度のチェック
3. 実行によるチェック

ソースコードのコンパイルは提出されたファイルがコンパイルできたかの可否を判断する．

そして，仕様の満足度は与えられたクラス図に記載されている状態（メンバー）や振る舞い（メソッド）が正しく実装されているかチェックを行う．また，そのときにインデントが正しく記述されていること，状態を振る舞いのなかで参照する際に this をつけて参照していることなどのコーディング規約をチェックする．

最後にプログラムを実行し，実行結果が正しく行われているかを判断する．従来はこの採点プロセスを複数の教員間で採点基準を統一するために 3 人の教員が受講者全員分を 1 回ずつ採点して調整を行っていた．この作業は採点時のミスを防ぐためでもある．採点の課程にあるコンパイルや実行はバッチファイルなどで自動化されていても，採点に関するチェックのほとんどが目視による手作業となっていた．

3.2 システムの目的

採点支援システムの目的は教員によって目視で行われる手作業の負担を減らし，システムを利用することで採点基準の統一化を図ることである．また，採点支援システムによって，より詳細な判断ができることを要望として教員側から挙げられた．

3.3 システムの構成

システムは図 2 のようなディレクトリ構造で教員ごとの学生のプログラムを管理し，採点支援を行う．

tests フォルダの中にあるフォルダが教員フォルダとして扱われ (teacher001 など)，その教員フォルダの中が各学生の提出したプログラムを管理するフォルダとして扱われる．したがって，このディレクトリ構造に従って配置すれば，教員や学生の数に対して柔軟に対応できるように設計されている．

テストの解答は answer フォルダの中に配置し，学生の解答に対する各チェック項目を判断するプログラムを check フォルダに作成する．check フォルダで作成するプログラムはシステムがライブラリとして提供する API を利用して作成される．

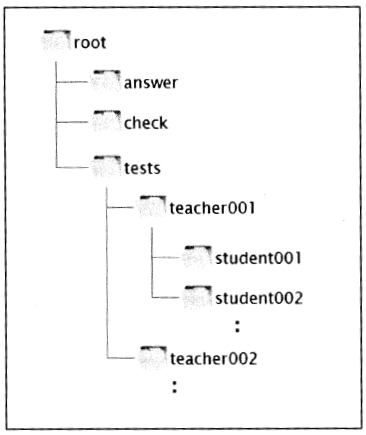


図 2: システムで扱うディレクトリ構造

3.4 システムの出力結果

各チェック項目を処理した結果は XML ファイルとして出力される．そして，この XML ファイルを読み込み，教員が採点の際に利用する各種ファイルへと出力する．図 3 はシステムが出力する各種ファイルである．

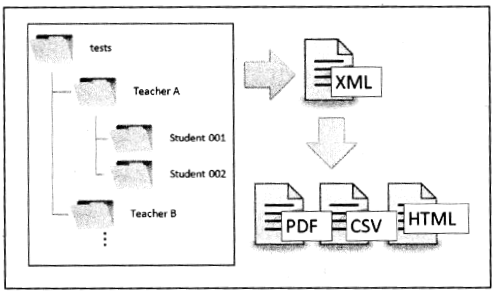


図 3: システムが出力するファイル

CSV ファイルは学生の点数を Excel で管理用，HTML ファイルはハイパーリンクを用いて学生の結果を表示するビューを提供し，PDF ファイルは教員が担当する学生のチェック結果を紙で出力するために用いられる．HTML のビューの例を図 4 に示す．

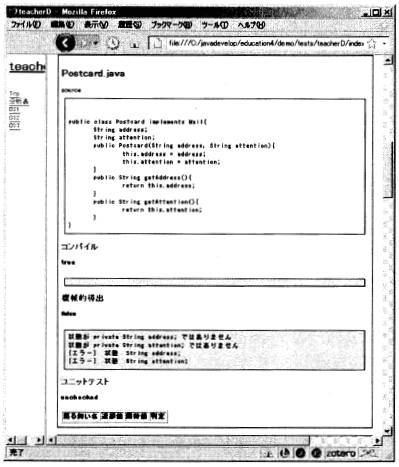


図 4: HTML のビュー

4. 採点支援システムの実行時における問題とその解決法

プログラムを評価する技法は従来の研究[4]により，プログラムに対して静的または動的なチェック方法があることが知られている．静的な方法はソースコードを解析するもの，動的な方法は実行した結果を比較するものである．学生から集められたプログラムはスクラッチから作成されたものであり，仕様を満たす範囲で自由にプログラムを作成することができる．しかし，作成されたプログラムにはあらゆる部分で静的・動的なチェックの際にエラーが発生する．したがって，エラーが発生した場合でも柔軟にそれらを加味した採点を行うために，開発システムには予測されるエラーなどに対処できるようにする必要がある．

今回，試作した採点支援システムは Java のバージョン 6 を用いて開発された．その設計・開発における各種チェック項目に対しての Java を用いた評価機構やエラーへの対処などの手法を紹介する．

4.1 コンパイル

学生から提出された各プログラムファイルは Java Compiler API を用いてコンパイルを行った．Compiler API を用いることにより，各ファイルをコンパイルする際にプロセスを立ち上げる必要がないため，システム実行時の OS への負荷を軽減し，処理の高速化が図れる．

また，コンパイルエラーに対してエラーの発生した行番号などの詳細エラーメッセージを解析することなく直接 API から取得できる．さらに，クラス名と保存したファイル名が違うエラーに対しては，学生が作成したファイルを編集することなく，正しいファイル名を割り当ててコンパイルすることができる．

4.2 仕様の満足度のチェック

学生の作成したプログラムがクラス図に記載されているとおりに作成されているかを構文解析技術によってチェックを行う．具体的には answer フォルダにある解答のプログラムと学生のプログラムを構文解析によってプログラムの抽象構文木を得る．そして，それらの抽象構文木を比較して正しく作成されたかを判断する．

4.3 重複したクラス名

提出されたプログラムが API 仕様に従って，正しい返却値を返すかを判断するためにユニットテストを行うとき，同じ名前のクラスが複数存在する状況での実行は正しく実行されない．1 つの解決策として学生それぞれのフォルダごとを対象としてプロセスを立ち上げてプログラムを実行する方法がある．しかし，この方法では同名のプログラムが 1 つしか実行できないので，解答と学生のプログラムの返却値を直接の比較することは不可能である．そこで，Java による解決策としてクラスローダを使用する方法がある．各学生のプログラムをそれぞれのクラスローダで管理し，実行することによって同じクラス名でもそれぞれ実装された内容で正しく実行される．さらに，解答のプログラムも同様に扱うことができるので，ユニットテストを期待した返却値をプログラムに入力して判断するのではなく，解答のプログラムと同様の返却値が得られるかを比較することが可能となる．

4.4 ユニットテスト

ユニットテストを実行する際に学生が作成した振る舞い名にスペルミスなどある場合，その振る舞い

の中身が正しいとしても呼び出す名前が違うのでユニットテストをすることができない．そこで，ユニットテストを実行するときにリフレクションを用いて行う手法を採った．リフレクションによる振る舞いの呼び出しは正規の呼び出しに比べて，パフォーマンスが悪くなるというデメリットが存在するが文字列によって柔軟に振る舞いを実行できるという特徴が利用できる．これにより，例えば学生が作成したプログラムの振る舞いにスペルミスがあったとしても，ユニットテストを行うことができる．

4.5 セキュリティ対策

Java でコンパイルされたファイル（バイトコード）は単体での実行はできないようになっており，JavaVM で実行する際に実行環境の制限をかけることができるセキュリティ機構がある．これを利用することで，採点支援システムを扱うマシンで安全に実行することができる．

5. 評価

今回の採点支援システムの試作ではコーディング規約のチェック以外はシステムによりチェックすることが可能となった．そして，平成 20 年度のコンピュータプログラミングＢの期末テストにてシステムを導入し，教員 2 名が試験運用し評価を行った．システムは提出された学生数 196 名の合計ファイル数 1371 個のプログラムファイルに対して，表 1 のスペックを持つマシンで実行したところ，処理が完了するまで 1 分 37 秒であった．教員 2 名による主観的な採点効率は 2 名とも採点業務の時間が約 3 分の 1 になったという評価が得られた．また，システムにより従来 3 名による採点業務が 2 名で可能となった．しかし，試験ごとにシステムの API を用いてチェックプログラムを書く必要があり，そのコード量はとても少ないとはいい難い量であった．

表 1: システムの評価環境

OS	Windows XP
CPU	Intel Core 2 Quad 2.4GHz
Memory	2GB

6. まとめ

本学部ではオブジェクト指向プログラミングの教育の一環としてコンピュータプログラミングＢの講義がある．その講義の試験は学生にクラス図と API 仕様を参照しながら時間内にスクラッチによってプログラムを完成させるものである．

本研究では提出されたプログラムの採点を支援するシステムの試作を行った．試作したシステムは約 1400 個のプログラムファイルを 2 分以内に処理することができる．また，システムを導入することで採点業務の時間が約 3 分の 1 に短縮されたという教員 2 名による主観的評価が得られた．

システムは Java で開発され，Java によるセキュリティーの高い，学生のプログラムに対するエラーに柔軟に対応できるものとなった．

しかし，採点項目のなかで実装されていないものや試験ごとに作成するコードの量などの課題が残される結果となり，今後はシステムを発展させシステムのみで採点業務のほとんどを行えるようにしていきたい．

参考文献

1) 藤原 巧，松浦 佐江子：評価方法に従った自動採点を可能にするプログラム採点支援ツール，電子情報通信学会技術研究報告，Vol.105，No 298，pp.23-28 (2005)
2) 小西 達裕，鈴木 浩之，伊東 幸広：プログラミング教育における教師支援のためのプログラム評価機構，電子情報通信学会論文誌，Vol.J83-D-I，No.6 pp.682-692 (2000)
3) 大澤 佑至，高岡 詠子：Java プログラミング単位認定型 e-Leaning 授業におけるプログラミングスキルに関する解析，情報処理学会研究報告，2008-CE-95，Vol.2008，No.64，pp.87-94 (2008)
4) 内藤 広志，斉藤 隆：プログラミング演習のための進捗モニタリングシステム，情報処理学会研究報告，2008-CE-93，Vol.2008，No.13，pp.33-40 (2008)
5) Sun Microsystems：Sun Developer Network，http://java.sun.com/