

# オブジェクト指向における段階的な開発スタイルへの誘導する 初級 Java 演習の支援システムの試作

辻健人<sup>†</sup> 富永浩之<sup>†</sup>

香川大学<sup>†</sup> 香川大学<sup>†</sup>

## 1. はじめに

大学情報系学科では、オブジェクト指向プログラミングとして、C++/Java 言語を対象とした入門授業が開講されている。そこでは、クラス継承、インスタンス生成、メソッド定義、カプセル化、イベント駆動、状態遷移、メッセージ通信など、必須な文法事項を習得する。

本研究では、それに続く、オブジェクト指向による設計と実装を実践的に行う演習を提案している[1]。対象者は、3年次程度とし、3~4人のグループで、チーム開発を体験させる。

本研究では、教育効果の高い問題設定と、必要機能を提供する演習支援システム TooDex を開発している[2]。本システムでは、実行性能テストと品質検証テストの2段階のテスト手法を用いて、オブジェクト指向開発の学習を支援する。

## 2. 本演習における外部評価の実行性能テスト

まず、プログラムの機能要件として、実行性能テストを行う。これは、外部評価であるプログラムの入出力サンプルでの実行結果を確認する。これを実行性能テストと呼ぶ。これには、個別と集結の2種類がある。

個別テストのレベルでは、満足すべき仕様や入出力サンプルを構築し、模範コードも用意しておく。設問に従い、メソッド単位で適当な箇所を幾つか隠蔽し、穴空きの出題コードとする。メンバ各自が担当した部分のコーディング作業を進め、個別に提出した解答コードで、模範コードの該当部分を置き換えて、実行する。

集結テストでは、チーム全体の解答コードを模範コード上で集結し、実行する。その際、メンバが解答したメソッドやクラスの中で、使用するものを決定し、1つのプログラムとする。個別テストでエラーが生じなくとも、全体の整合性が満たされなければエラーが生じたり、実行結果が正しくない状態が発生する。

ここでは、対象となるデータの生成、共通処理の補助メソッドの作成と検証、オブジェクトの責務と概略設計、部分仕様に基づいた段階的な実装、各種のサンプルによる結合テストを、より実践的に体感させる。また、非機能要件としての処理速度や応答性など、パフォーマンスも測定する。

## 3. 本演習における内部評価の品質検証テスト

さらに、完成したプログラムの非機能要件の、システムによる自動的な検証や支援を導入していく。このような実装の適切さを確認するテストを品質検証テストとする。ここでは、リファクタリングを意識したり、チームの解答コードと試練コードを結合し、試練プログラムを実行して評価する。試練コードは、テスト駆動開発における、テストコードおよびその拡張である。例えば、カプセル化について、オブジェクト変数へのアクセス修飾子を正しく設定しているかどうか、試練コードからの攻撃的なアクセスで確認する。また、継承について、下位クラスが、与えられた上位クラスから正しく派生されているかどうか確認する。コードの冗長性や脆弱性を突いたり、拡張性や再利用性の度合いを測ったりする攻撃的なテストに挑戦させる。

## 4. 整列算法を事例とする演習事項

本研究では、典型的な事例として、各種の整列算法を採用する[3]。ただし、個々の算法そのものは既習とし、開発手法の題材として扱う。したがって、C言語でのプログラムは予め提示しておき、オブジェクト指向プログラミングの観点から、Java言語に移植する例題となっている。

まず、整数のデータに限定し、データ構造を定義する。データ列の生成として、整列済、乱数列、入力指定の3通りのメソッドを実装する。整列済は、結果の検証に用いる。ここで、抽象クラスの具体化、上位クラスからの継承による派生クラスの生成などを実習する。

次に、基本処理として、データの大小判定、データ列の要素の比較と交換、指定範囲の部分列の整列済の判定などを実装する。ここで、結合度を抑えて凝集度を高めるモジュール化、オブジェクトと責務の分担を意識させる。

A Trial Version of a Support System for Applied Java Programming Exercise toward Stepwise Development in Object-Oriented Style

<sup>†</sup>Kento TSUJI, Kagawa University

<sup>†</sup>Hiroiyuki TOMINAGA, Kagawa University

続いて、3つの単純整列法(交換・選択・挿入)における二重反復の内側の処理を、指定範囲に対する独立したメソッドとして実装する。各メソッドは、簡単なテストコードから試行的に呼び出し、単体テストで確認する。最後に、各処理を実装し、整列算法としての結合テストを行う。単なる結果だけでなく、途中の過程も確認の対象である。

課題の発展として、逆順での整列、部分的な整列、対象のデータ型の変更などに取り組む。性能評価として、比較や交換の回数の出力、処理時間の計測などもコードに組み込む。ここでは、アサーションの導入、再帰法での再実装、アクセス修飾子による変数の可視性の適切さも確認する。

### 5. 支援システムの機能と演習の進行

本演習は、次の4つのモデルから構成される。それぞれ、科目、設問、大問、小問である。科目には授業の概要や開催期間を設定する。設問は科目と対応しており、設問は、学生が演習に取り組んだ際に成し遂げる最終目標となる。例えば、整列算法をテーマとして扱う場合「整列算法を実装せよ」が設問に該当する。大問は、学生が取り組む際の段階的な目標である。整列算法において、データの生成、ソート、出力などが大問に該当する。小問は、大問をさらに詳細化、分割化したものである。整列算法におけるデータの生成という大問では、「標準入力からデータ列を生成」や「ファイルからデータ列を生成」が小問に該当する。

学生の演習の前半の手順について説明する(図1)。演習を行う際は、教員の指導に従ってチームは大問を選択する。チーム内でプログラムの得意、不得意などを考慮し、学生は大問から小問を選択する。それぞれ学生は、担当した小問を実装する。また、提出に応じて個別テストを実行する。個別テストは、実行性能テストと品質検証テストの両方を含む。テスト実行の際は、チーム内のコードは補完せず、常に模範コードをマージしてテストを行う。この時点では、テストはすべて通過する必要がなく、最低限のテストが通れば良しとした包括方式による評価を行う。チーム内の全メンバが、最低限のテストを通過すれば、チームの代表が小問の完了を行う。

小問が完了したら、チーム全員で大問に取り組む(図2)。大問の着手を開始した時点で、システムは、チームのマージ済みコードに対して、集結テストの実行とメトリクスの測定を行う。学生は、その結果をみてチーム内でディスカッ

ションを行い、問題箇所を考えコードの改善を行う。改善するための提出があるたびに再度テストとメトリクスを実行し、結果を更新する。集結テストでは、個別テストと異なり、通過したテストが多ければ多いほど高い評価をつける加算方式を採用する。

### 6. おわりに

オブジェクト指向によるチーム開発を意識させる初級 Java プログラミングの演習支援システムを提案している。模範コードの一部を隠蔽して設問として出題し、チームで分担して解答する。各メンバの個別テストの後、チーム全体での集結テストを行う。さらに、試練コードによる検証テストを行う。これは、非機能要件であるコードの品質を確認する。現在は機能検討の段階であり、開発を進めていく。問題も構築する。

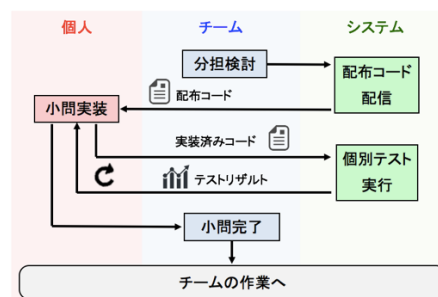


図1 個人の着手順

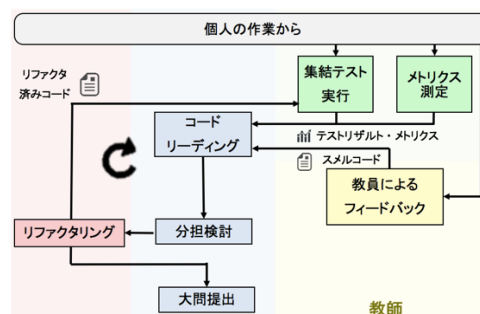


図2 チームの着手順

### 参考文献

- 1) 玄馬史也, 富永浩之: オブジェクト指向の入門的な開発スタイルを意識させる初級 Java プログラミングのグループ演習の構想, 情報処理教育シンポジウム SSS2015, Vol.2015, No.9, pp.1-6 (2015).
- 2) 辻健人, 玄馬史也, 富永浩之: オブジェクト指向の入門的な開発スタイルを意識させる初級 Java プログラミングのグループ演習の支援システムの GUI 検討, 情報処理学会 第 79 回全国大会, Vol.79, pp.649-650 (2017).
- 3) 辻健人, 玄馬史也, 富永浩之: オブジェクト指向における段階的開発への誘導を目指す初級 Java 演習支援の課題検討, 教育システム情報学会 第 40 回全国大会, Vol.40, pp.331-332 (2017).