

段階的詳細化によるオブジェクト指向開発を誘導する

Java 演習支援システムの出題採点機能の実装

石井怜央^{†1} 辻健人^{†2} 富永浩之^{†3}香川大学^{†1} 香川大学^{†2} 香川大学^{†3}

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、メソッド定義などの、基本的な文法を習得する。しかし、これらの授業で扱う課題は、単なる文法の確認課題となりがちで、オブジェクト指向らしい表現まで扱うことは難しい。その結果、手続き型言語のような処理の記述に留まってしまいう学生も多い。

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体験する演習を提案している。問題を構築し、演習支援システム TooDex(図 1)を開発している[1][2]。継続的にコードを改良することで、各オブジェクトの責務を意識させる演習を目指す。関連研究として、学生の解答を、テストを用いて自動的に評価するシステム ArTEMiS がある[3]。本研究では、より初級的な学生を対象とした教育支援を目指す。

本論では、支援システムにおける、解答の提出および採点機能について述べる。学生の提出した解答に対し、テストケースを実行し、採点結果をフィードバックする。

2. OOP 開発を意識した Java 演習の概要

本演習では、模範コードの一部を隠蔽した穴埋め形式の設問として出題し、チームで分担して解答する。段階的な設問により、継続的インテグレーションやリファクタリングを誘導する。オブジェクトの責務を意識させ、1つの目的のために振舞いのみを実現することを実践させる。また、達成度や理解度を確認するため、システム上で実行性能テストと品質検証テストを実行する。これらのテストには、チームの各メンバーが分担したコード部分のみを模範コードに差し替えて実行する個別テストと、チーム全体でコードを統合して実行する集結テストがある。個

別テストと集結テストはそれぞれ、実際の現場における単体テスト、結合テストに近いものである。

実行性能テストは、ブラックボックスへの機能要件としての外部評価である。入出力サンプルと実行結果の照合や処理速度などを検証する。

品質検証テストは、非機能要件であるソースコードの品質を注目した内部評価である。ソフトウェアメトリクスの計測や試練コードで、凝集度や結合度、コードの安全性や拡張性などを検証する。ソフトウェアメトリクスによる静的解析で、冗長性や複雑度が異常な値を示すコードには警告を与える。また、試練コードでは、`private`に定義されるべきメソッドなどに、あえてアクセスすることで、エラーが出るかなどもテストする。

これらにより、必要な情報のみの参照や開示を行い、他のオブジェクトとのメッセージ通信を最小限に留めることを意識させる。

3. 演習支援システム TooDex

本演習は、科目、設問、大問、小問の4つのモデルから構成される。科目は、授業としての科目と対応しており、授業の概要や開催期間を設定する。設問は、学生が演習に取り組んだ際の最終目標であり、開発現場におけるプロジェクトである。例えば、「代表的な整列算法の実装」「三目並べの実装」などが挙げられる。大問は、設問を達成するための、段階的な目標である。整列算法における、データの生成、ソート、出力や、三目並べにおける、盤面の出力、勝敗判定などが該当する。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、「標準入力によるデータ列の生成」や「ファイル読み込みによるデータ列の生成」などが該当する。学生は、各大問における、各小問の実装とリファクタリングを繰り返す。全ての実装が完了すれば、次の大問に移る(図 2)。

小問は個別テスト、大問は集結テストがそれぞれ対応する。小問解答時には、常に模範コードに対してマージし、個別テストを行う。小問を全て完了したら、これまでの小問に対する解

Implementation of Questioning and Scoring Functions in a Support System of Java Programming Exercises for Object-Oriented Development by Stepwise Refinement.

^{†1}Reo ISHII, Kagawa University

^{†2}Kento TSUJI, Kagawa University

^{†3}Hiroyuki TOMINAGA, Kagawa University

答全体を模範コードにマージして集結テストを行う。そのため、未実装の部分があっても、ブラックボックス化されたメソッドとして、外部仕様のみを用いて、実装を進めることができる。小問解答時に、前段までの設問の実装に詰まった場合や、チームで分担した場合にも、解答を続けることができ、チーム開発における、分担や委託の体験のシミュレーションにもなる。

また、解答は全て穴埋め形式で行うが、穴抜き箇所の指定は、出題に用いる模範コード中に、コメントとしてタグを埋め込み、指定する。現在は、開始タグとして“// !["，終了タグとして“// !]" を利用している。支援システム側で、タグを識別し、指定箇所を隠蔽して表示する。

4. TooDex における採点機能

本演習は、Java 言語の利用を想定している。そのため、テストツールである JUnit を用いて、学生の解答を評価するためのテストを作成している。また、ビルドとテストの自動化のため、ビルドシステムである Gradle を採用している。それらの実行環境として、並列処理や安全性、動作速度などを考慮し、コンテナ型仮想環境 Docker を利用している(図 3)。

学生の解答に対する、システムの動作は以下のようなになる。まず、学生は、各小問について、TooDex の Web 画面上で、穴抜き部分を埋めて提出する。システム側では、模範コードの該当部分を、提出されたコードに差し替えて記録する。その後、Docker コンテナを起動し、解答コードやテストなどが含まれる、プロジェクトのディレクトリをマウントする。コンテナ内で Gradle を用いてプロジェクトをビルドし、テストを実行する。テスト結果は、Gradle により、xml 形式で記録される。システム側で、それらをパースし、学生に Web 画面上に表示してフィードバックする。この時、いくつかのテストには、事前にメッセージを設定し、テストに失敗した時のヒントとして表示する。

5. おわりに

本研究では、段階的な開発スタイルを誘導する中級 Java 演習を提案し、支援システム TooDex の開発を行っている。TooDex は、部分的に完成したコードの提出に対し、模範コードの該当部分に差し替え、各種のテストを実行する。

段階的にテストケースを満足するように、継続的にコードを改良することで、各オブジェクトの責務を意識させる演習を目指している。

本論では、学生の提出した解答に対し、テストケースを実行し、採点結果をフィードバックする機能を実装した。ビルドシステムとして

Gradle を利用しており、JUnit のアサーションを用いて作成されたテストを実行する。これらは、スケーラビリティなども考慮し、Docker コンテナ内で実行する。テスト結果は、Gradle により作成された xml をパースし、事前に設定したコメントなどと合わせて、Web 画面上に表示することでフィードバックする。今後は、残りの機能を実装、問題の構築を行い、試行実践を行う。

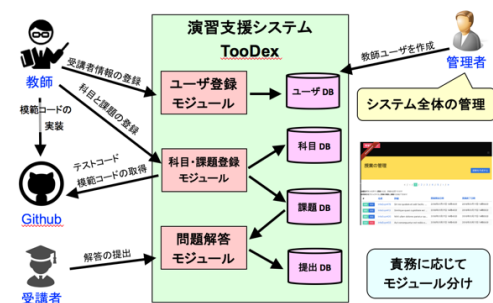


図 1 演習支援システム TooDex

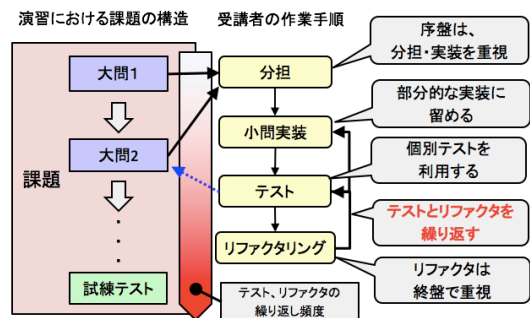


図 2 受講者の演習の手順

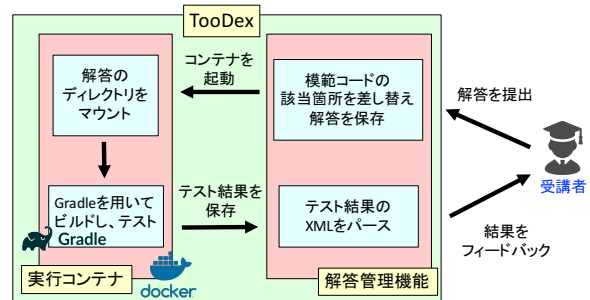


図 3 TooDex の採点時の流れ

参考文献

- 1) 辻健人, 富永浩之: 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, 情報処理学会 第 80 回全国大会, Vol.80, No.1, pp.817-818 (2018).
- 2) 石井怜央, 辻健人, 富永浩之: オブジェクト指向における段階的な開発とテスト駆動を誘導する応用 Java 演習の支援システム ~ 整列算法のコードの修正と拡張を題材とする演習進行の仮想実験とユーザ評価 ~, 信学技報, Vol.118, No.46, pp.11-16 (2018.05).
- 3) Krusche, S., Seitz, A.: ArTEMiS - An automatic assessment management system for interactive learning, Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Vol.40, pp.284-289 (2018).