

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

辻健人[†] 玄馬史也[†] 富永浩之[†]

香川大学[†] 香川大学[†] 香川大学[†]

1. はじめに

大学情報系学科では、初年次の初級プログラミング演習として、手続型のC言語を採用することが多い。ここで、基礎的な文法事項、データ構造と基本的なアルゴリズム、要件定義とウォーターフォール開発を習得する。続いて、C++/Java言語を対象とし、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、オブジェクト指向に必須な文法事項を習得する。また、簡単なクラス設計、スパイラル開発を体験する。その先には、デザインパターン、リファクタリング、責務の委譲、テスト駆動など、コードの品質を意識した、より実践的なプログラミングの技能が求められる。

2. 本提案の OOP 開発演習の教育目標

本研究では、初級 Java のプログラミング演習の支援システムを提案する[1][2]。対象者は、情報系学科の3年次程度とし、3~4人でグループで、チーム開発を体験させる。文法事項というより、オブジェクト指向プログラミングのコーディングスタイルを意識させる。

このようなアプローチとしては、教授者による直接的な指導、学習者同士の相互評価の比重が大きい。本研究では、教育効果の高い問題設定や必要機能を議論し、支援システムの開発を目指す。まずは、プログラムの機能要件の外部評価である入出力サンプルでの実行結果の確認を実装する。次に、その拡張として、非機能要件の内部評価に、システムによる自動的な検証や支援を導入していく。

3. 本提案の OOP 開発演習の出題構成と意図

本提案の演習では、コード部分の穴埋め方式による、ソースコードの共同作成としてのプログラミング課題を出題する。設問は、メソッドやクラス単位で分割されており、チームのメン

バは、分担してコーディングを行っていく。教師は、事前に、満たすべき仕様や入出力サンプルを構築し、模範コードを作成しておく。設問に従い、適当な箇所を隠蔽して穴空きの出題コードとする(図 1)。ここで、メソッド定義ごとに、「コードを全て明示」「変数の定義やヒントのみを公開」「概略としてスケルトンコードのみを公開」「メソッドの外部仕様のみを公開」「存在を完全に隠蔽」の種別を設けておく。これに基づき、模範コードの一部または全てを隠蔽した出題コードを、チームに公開する。

演習は、前半の外部評価と後半の内部評価に分ける。前半では、与えられた仕様に従い、設問ごとに基礎的な内容のコーディングに取り組む。プログラムの外部評価として、入出力データによる実行結果の照合で成果を判定する。後半では、コードの品質も意識し、リファクタリングを行ったり、コードの修正や拡張に取り組む。プログラムの内部評価として、各種のテストで判定する。

4. 外部評価としての個別テストと集結テスト

前半の外部評価は、個別と集結の2種類の実行テストで行う。個別テストでは、メンバ各自が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差し込み、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。

本提案でのグループ演習は、図2のような流れで進行する。学生は、教師からの課題提示に対し、チーム内で相談して作業分担を決め、リーダーが各自に指示を出す。課題によっては、データ型を揃えるなどの全体の方針も定める。各メンバがコーディング作業を進め、個別テストに至る。個別テストでは、メソッドやクラス単位でコードを提出し、模範コードと結合して実行することで入出力の検証を行う。

個別テストの結果、文法事項やエラーなど、個人作業で生じた問題点をチーム内で互いに助言し合う。ここでは「分かる人が分からない人に教える」という1対1のコミュニケーションが中心となる。チーム内で解決できない時は、教員に質問し、回答を得る。

GUI Design and Mockup of a Support Tool for Group Exercises of Preliminary Java Programming for Introduction to Object-Oriented Developing Style

[†]Kento TSUJI, Kagawa University

[†]Fumiya GEMBA, Kagawa University

[†]Hiroyuki TOMINAGA, Kagawa University

各々の担当した部分において、個別テストが終了した場合、メンバ全員のコードを合わせて集結テストを行う。集結テストでは、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したりする。ここでは、各自の解答コードを全員で共有し、コードレビューを行う。チーム全体で議論を行い、リーダーが意見を集約して、最終的なプログラムを完成させる。教員は、各チームを巡回し、適切なアドバイスを与える。

図3は、モックアップのGUIにおけるWebページの設問一覧タブである。左側に設問ごとの担当ボタンがあり、各メンバが押して主担当を宣言する。状況に応じて、複数担当とし、補助者を追加することもできる。中央は、模範コードの一部を隠蔽した出題コードである。担当分の解答コードをテキストボックスに挿入して、送信すると、個別テストが実施される。右側は、チャット欄である。必要に応じて、担当表のモーダルウィンドウが現れる。

5. 内部評価としての試練コードでの検証テスト

後半の内部評価では、実装の適切さを確認するため、検証テストに挑戦する。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。正誤だけではなく、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。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。

課題の総合評価としては、チームの共同作業の内容を、個人テスト、集結テスト、検証テストにより、内外から評価する。また、教師は、最終的なプログラムを目視で確認し、採点する。その他にも、プログラムに関する理解度を図るために、各グループに対してコードレビューや口頭試問などを行う。学生は、教員からの指摘を受けるだけでなく、他のグループと議論し、互いに問題点や良かった点を挙げ、グループ外からも幅広い知識を得る。

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

ムを提案した。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。

参考文献

- 1) 稲見望, 富永浩之: プログラミング演習のためのグループチャット型 CSCL, 信学技報, Vol.101, No.706, pp.107-114 (2002).
- 2) 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 SSS, Vo.2015, No.9, pp.1-6 (2015).

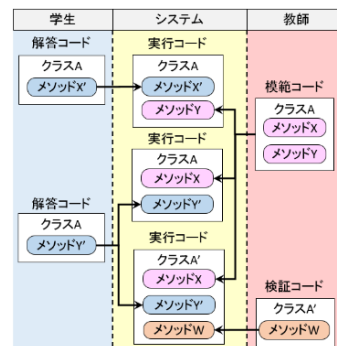


図1 チーム解答の各種のテスト

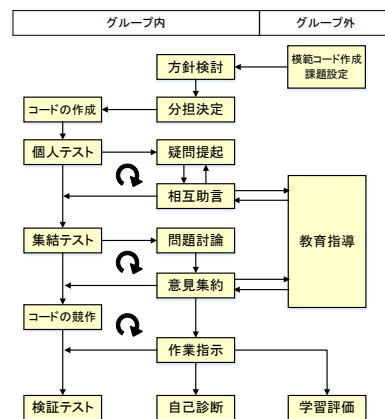


図2 グループ演習の進行モデル

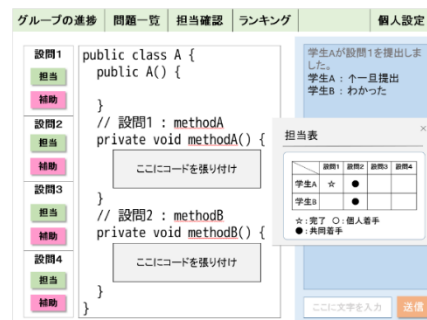


図3 設問一覧ページ