

## 段階的詳細化プロセスによる LLM を活用した プログラム論理構成・エラー解決支援

### Program Logic Structuring and Error Resolution Support Using LLMs Through a Stepwise Refinement Process

原田 裕太<sup>\*1</sup>, 中村 勝一<sup>\*2</sup>, 宮寺 庸造<sup>\*1</sup>  
Yuta HARADA<sup>\*1</sup>, Shoichi NAKAMURA<sup>\*2</sup>, Youzou MIYADERA<sup>\*1</sup>

<sup>\*1</sup> 東京学芸大学

<sup>\*1</sup> Tokyo Gakugei University

<sup>\*2</sup> 福島大学

<sup>\*2</sup> Fukushima University

Email: m248120f@st.u-gakugei.ac.jp, nakamura@sss.fukushima-u.ac.jp, miyadera@u-gakugei.ac.jp,

**あらまし**：本稿では、大学のプログラミング初学者が課題の理解から構想、実装、検証に至るまでの一連の思考プロセスにおける問題に着目し、段階的詳細化に基づく 5 つの思考プロセスを用いた学習モデルとその支援システムを提案する。生成 AI を活用し、学習者が段階ごとに考えるべきポイントを明確にし、ながらプログラムを構築できる環境を目指す。

**キーワード**：プログラミング教育、段階的詳細化、学習支援モデル、LLM によるアドバイス生成

#### 1. はじめに

大学におけるプログラミング演習授業では、初学者が課題文を読んでも「何から始めればよいのか分からない」と感じ、十分な構想を持たないまま実装に着手して行き詰まるケースが多く見受けられる。課題の要件を適切に分析・整理しないままプログラムを記述することで、そもそも意図されたロジック自体が論理的に破綻している(問題点 1)場合のみならず、処理の意図とプログラムの挙動との間に齟齬が生じる(問題点 2)場合も多く、結果として論理的誤りが顕在化しやすくなる。また、学習者一人ひとりのつまずきのパターンは多様であり、個別の状況に応じた適切な支援を提供することが困難である。さらに、学習者は自身がどの段階でつまずいているかを客観的に把握することが難しく(問題点 3)、適切な支援を求めることすら困難な状況に陥りやすい。

これらの問題の解決には、課題の目的や要件を正しく把握し、全体の流れを捉えたうえで徐々に細部を詰めていく「段階的詳細化 (stepwise refinement)」の学習が有効であると考えられる。従来のプログラミング教育では、段階的詳細化の理論的有用性は早くから認識されていた<sup>(1)</sup>ものの、教育現場における体系的な指導法は確立されておらず(問題点 4)、多くの場合、学習者の試行錯誤に委ねられていた。

近年では生成 AI をはじめとする人工知能技術の教育応用が注目されている。たとえば、学習者のコードを自動で評価・採点するシステム<sup>(2)</sup>や、対話的にバグの原因や修正方法などのアドバイスを提示するシステム<sup>(3)</sup>などが開発され、実装フェーズにおける支援環境は着実に整備されつつある。しかし、これらの既存支援システムの多くは、コードの正誤判定やバグの指摘といった「実装後」の支援に主眼が置かれている。また、構想段階を支援するシステムであっても、フローチャートや疑似コードといった中間表現を介するものが多く、初学者にとってはコードを書く前にもう一つ別の作業が必要となり、かえ

って学習の負担が増してしまう。加えて、再帰関数のような概念ではフローチャートなどに表現すること自体が困難であることや、コードとの乖離が大きくなることも負担を増やす原因と考えられる。そのため、課題の理解から構想、実装、検証に至る一連の思考プロセスを自然な流れで体系的に支援することが求められている。

そこで本研究では、プログラミング初学者が課題の理解、構想・実装、検証に至る一連の思考プロセスを習得する支援を目的とする。本稿では、思考プロセスの体系化と LLM によるプロセスの支援システムの提案を行う。

#### 2. 思考プロセスの体系化

##### 2.1 概要

本章では、課題の理解、構想・実装、検証に至る一連の思考プロセスを体系化し、問題点 4 の解決を図る。構想と実装に関しては、段階に分離せず、プログラムの目的や処理の流れを考えながら段階的に詳細化し、その都度コードに反映する一体的な方法を提案する。検証においては、完成したコードに対して期待される出力が得られるかを確認し、エラーが生じた場合には実行過程を追跡しながら各変数の変化を辿ることで原因を特定し、処理の流れや妥当性を検証する。これにより、初学者が課題を論理的に捉え、自力で解決できるようになることを目指す。

本プロセスが対象とするのは、基本的なプログラミングの構文(変数、条件分岐、反復処理等)は理解しているものの、課題の理解、構想、実装、検証のいずれかの段階で困難を抱えるプログラミング初学者である。具体的には、大学初年度レベルのプログラミング初学者を想定している。

##### 2.2 プロセスの体系化

本研究では、新開による先行研究<sup>(4)</sup>を参考に、プログラミング初学者が課題に取り組む際の思考プロセスを、「①：理解」「②～④：構想・実装」「⑤：検証」の 3 段階 5 プロセスに整理する。それぞれのプ

プロセスの特性に応じた支援を目的として体系化する。

① 課題の理解(理解)

課題において、目的を理解し、「何を入力として受け取り、何を出力すべきか」を明確にする。

② 処理の大枠の決定(構想・実装)

課題の要件に基づき、実装すべき処理をトップレベルの手順に分割し、全体像を把握する。

③ 処理の詳細化(構想・実装)

決めた主手順について、それぞれの処理に必要な変数を明確にしなが、構造化プログラミングの基本制御構造（順次・選択・反復）に沿って処理の流れを詳細化する。

④ 条件・演算の具体化(構想・実装)

詳細かされた構造に対して、必要な条件式や計算式を記述し、実装可能な処理に仕上げる。

⑤ コードの整合性判定(検証)

完成したコードに対して、期待される出力が得られるかを確認する。エラー時には、実行過程を追跡し、各変数の変化を辿ることで原因を特定し、処理の流れや妥当性を検証する。

「②～④：構想・実装」に関しては、段階的に詳細化を繰り返し行うことを想定している。

3. 支援システムの概要

本章では、第2章で示した3段階5プロセスからなる思考プロセスに基づくアドバイス生成システムについて述べる。本システムは、構想を終えてから実装に移るのではなく、プログラムの目的や処理の流れを段階的に明確化しつつ、その都度コードに反映していく一体的な取り組みにより、初学者が自分の理解を可視化し、段階的に確かめながら学習を進められる環境を目指す。これにより問題点1～3の解決を図る。

3.1 プロセス別支援内容

支援システムは、思考プロセスの各段階において学習者が適切に次のステップへ進めるように促すことを目的とする。そのために以下の支援を提供する。

① 課題の理解

学習者は課題の目的を理解し、入出力の要件を正確に把握することを目指す。システムは課題文を解析して入出力明確化のための質問を生成し、理解度確認を行う。

② 処理の大枠の決定

学習者は課題要件から解法の全体的な方針を決定して、コード内に処理の大枠をコメント形式で記述することを目指す。システムは方針の妥当性を判定し、論理的整合性を確認し、学習者にアドバイスを提供する。

③ 処理の詳細化

学習者は各主要手順を基本制御構造で具体化し、必要な変数を適切に設計して、処理の具体的な流れを構築することを目指す。システムは制御構造や変数設計、処理順序の妥当性を確認し、抜け漏れなどが検出した場合には適切なナビゲーションを提供する。

④ 条件・演算の具体化

学習者は境界条件や特殊ケースを考慮した条件式・演算式を記述し、実行可能なコードとして完成させることを目指す。システムは式の正確性を確認

し、エラーを検出した場合には修正アドバイスを提供する。

⑤ 検証

学習者は期待される動作との整合性を確認し、エラーの原因を特定して適切に修正し、コード完成を目指す。システムは実行過程の追跡と変数値の変遷を辿るように促し、学習者が原因を特定し修正できるよう方針を提案する。

3.2 システム構成

本システムは、以下の4つの機能から構成される。システムの想定画面を図1に示す。

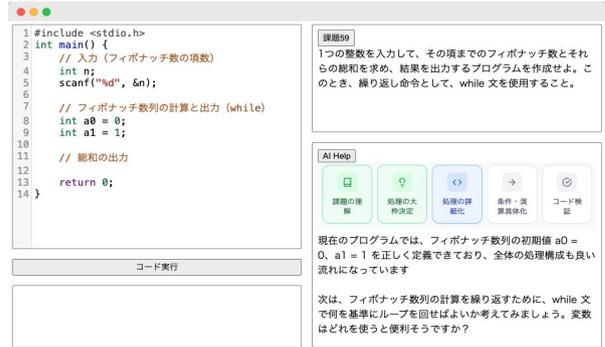


図1 支援システムの想定画面

(1) コードデバッグ環境 (左側)

実際のコードを記述し、実行結果を確認することができる。

(2) 課題内容表示機能 (右上)

現在の課題内容を確認することができる。

(3) 学習プロセス可視化機能 (右中央)

学習者の現在の思考プロセス全体の進捗を把握することができる。

(4) アドバイス生成機能 (右下)

学習者が「AI Help」ボタンを押すことによって、システムは学習者の現在のコードと3段階5プロセスの思考プロセスに応じたアドバイスを提供する。

4. おわりに

本研究では、プログラミング初学者の学習支援を目的として、課題の目的や処理の流れを段階的に明確化しつつ、その都度コードに反映していく一体的な取り組みを可能とする対話型支援システムを提案した。思考プロセスを3段階5プロセスに体系化し、課題理解から検証までを自然言語による対話を通じて段階的に支援することで、初学者が自分の理解を可視化し、確かめながら論理的に課題を解決できるようになることを目指す。

今後は、システムの実装と教育現場における有効性の検証を進めていく予定である。

参考文献

(1) 木村優那 他, “段階的詳細化能力とプログラミング学習の関係に関する研究”, 2015 年度人工知能学会全国大会, 2015.  
 (2) 関口祐豊 他, “PP-Checker 1.5: LLM との協働により標準出力を含むプログラムを評価する半自動採点システム”, インタラクシオン 2025 論文集, No.2A-14, pp.676-681, 2025.  
 (3) 上川貴之 他, “大規模言語モデルが生成した論理エラー修正コードと授業資料を活用したプログラミング支援”, DEIM2024, No.T5-A-4-03, 2024.  
 (4) 新開純子 他, “プロセスを重視したプログラミング教育支援システムの開発”, 日本教育工学科論文誌, ver. 31, pp. 45-48, 2007.