

芦屋大学論叢 第83号
(令和7年3月21日)抜刷

《研究ノート》

中学校技術科のプログラミング教育で登場する
デバッグ手法の教科書調査

岩 城 賢 一
藤 本 光 司
野 口 聡

《研究ノート》

中学校技術科のプログラミング教育で登場する デバッグ手法の教科書調査

岩 城 賢 一 (1)

藤 本 光 司 (2)

野 口 聡 (3)

(1) 芦屋大学大学院博士前期課程

(2) 芦屋大学大学院教育学研究科教授

(3) 芦屋大学大学院教育学研究科講師

1. はじめに

中学校技術科では「加工と材料の技術」(内容 A)、「生物育成の技術」(内容 B)、「エネルギー変換の技術」(内容 C)、「情報の技術」(内容 D) の 4 つの異なる技術を扱っている。その内、本稿で取り上げる「情報の技術」は比較的新しい内容である。「情報の技術」の内容は昨今の急速なデジタル社会への対応として今後も充実していくと予想する。その「情報の技術」のプログラミング教育は“問題を見いだして課題を設定し解決する力の育成”をねらいとした教育であると平成 29 年告示の学習指導要領 (2017)^{1),2)} に示されている。

ただ、そのプログラミング教育には課題がある。みんなのコード (2021)³⁾ の調査によると教員の内、35.3%がプログラミング教育を苦手と回答している。また、指導する際の課題として「ネットワークを利用した双方向性のあるコンテンツのプログラミング」および「計測・制御のプログラミング」にて「指導・授業の難しさ」および「教員の専門性の不足」が 50%以上を占めている。また、生徒についても、上記調査⁴⁾の「生徒の反応」では“熱中して突き詰める生徒と完全に諦めて分からないという生徒の二極化に近い。”と言った難しさについての回答があった。教員がプログラミング教育を苦手とする理由として楠見ら (2020)⁵⁾ の調査よりプログラミングの経験不足が挙げられる。その調査によると、中学校教員が大学生時代に“コンピュータのプログラミング”の授業を履修していた者は 26%と少なく、コンピュータを“プログラミング”のために利用する者は 8%と非常に少ないことが分かっている。さらにプログラミング教育の難しさの要因については小泉、藤川、渡壁ら(2017)⁶⁾ が調査を行っており、その調査の結果および考察にて、「プログラミングが難しい」と述べた生徒は「コンピュータが扱える機構の役割」や「センサ情報」についての理解が不足していると指摘している。

以上より、プログラミング教育の課題である苦手意識や難しさは、プログラミングした結果と動作の関係性、またはセンサなどのハードウェアの扱われ方の理解不足を原因とする、「どのようにプログラミングすれば目的の動作結果に至るのかが分からない」という思考によって引き起こされるのではないかと仮説を立てた。その対策としてデバッグ手法を身に付けることができれば、コンピュータの状態が把握できるため、プログラミング教育の難しさが軽減され、苦手意識も薄まるのではないかと予想した。

そこで本調査では、中学校のプログラミング教育で主に使用されている検定教科書にデバッグ手法が紹介されているかを調べた。

2. デバッグ（間違いのあるプログラムの修正）の概要

この章ではデバッグの概要を紹介する。デバッグはバグ（間違い）を修正する作業を指しており、そのバグにも大きく2種類存在する。1つ目は「構文エラー」と呼ばれ、プログラミングの記述ミスによりソフトウェアとして実行できない状態を指す。2つ目は「論理エラー」と呼ばれ、ソフトウェアとして実行できるものの、その実行結果が設計した通り、または思惑通りにならない状態を指す。いずれにしても修正が完了するまでの流れは①「バグの検出」、②「原因の特定」、③「プログラムの修正」の順に進み、間違いが解消されるまで繰り返すことは同じであるが、“どのように①「バグの検出」をし、②「原因の特定」をするか”という方法が異なる。具体的には次節の各修正過程にて、上記の①～③の流れに沿って説明する。

なお、論理エラーは構文エラーよりも修正することが難しい。構文エラーは「プログラミング言語のコンパイラ⁷⁾の判定」と「プログラミングした結果」の差異によって発生する。コンパイラの判定は不変であるため、コンパイラの指摘に従いプログラムを修正すればバグはなくなる（図1参照）。一方、論理エラーは「人が想定する動作（設計資料など）」と「プログラミングした結果を含むコンピュータの動作結果」の差異により発生する。どちらも状況により変化するため、どちらを修正すれば適切かという判断が事例ごとに異なってしまう（図2参照）。

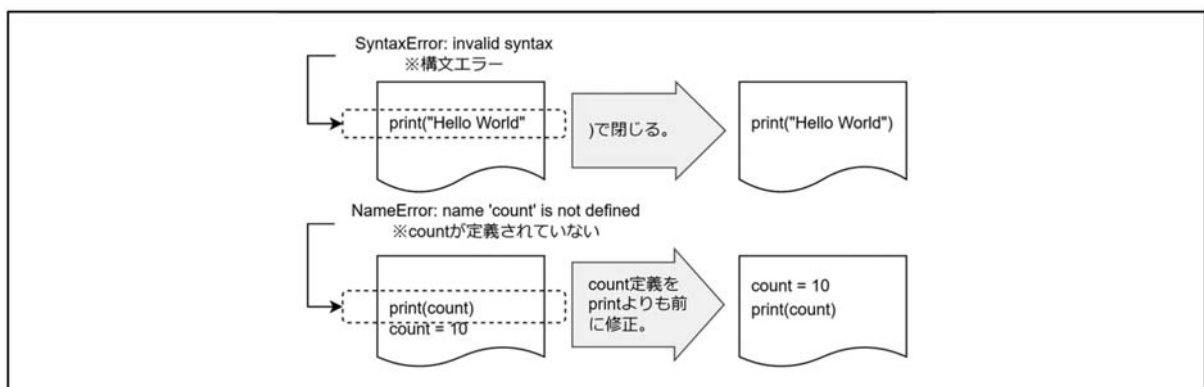


図1 構文エラーの修正のイメージ

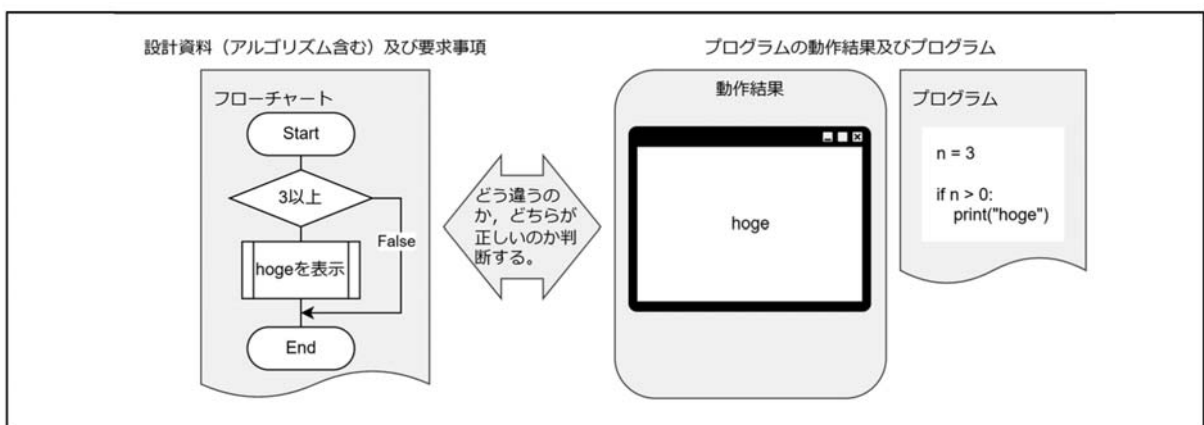


図2 論理エラーの修正のイメージ

2.1 構文エラーの修正過程

構文エラーはコンパイラが検出するエラーである。コンパイラからは①②「エラー内容（発生原因）とファイル名と行番号（発生箇所）が通知される」。その箇所をコンパイラの仕様に沿うようにプログラムの修正を行う。ただし、当該箇所より上流のプログラムにエラーの原因があり、当該箇所を修正してもバグが解消しない場合がある。その場合はコンパイラの処理がプログラムの上流から下流に向かって処理することを踏まえ、②「発生箇所から上流に遡って原因を特定」していく。原因の特定後、③「ソースコード（プログラムを記述したファイル）の一部を変更（削除も含む）しながら、コンパイルを繰り返す」ことを行う。

なお、最近では ChatGPT を用いて構文エラーを修正させることができる。宇都宮、秋山、近藤、亀井、鶴林（2023）⁷⁾ の評価では 70%以上確率でコンパイルに成功できたことを確認している。また、そもそも構文エラーが発生しないブロック型のプログラミング言語（Scratch⁸⁾）など）も存在している。

2.2 論理エラーの修正過程

論理エラーは設計者またはユーザーなどの人が検出するエラーである。なお、前提として構文エラーは全て修正済みであり、コンピュータがプログラムを実行できる状態になっている。そのため、コンピュータとしてはプログラミングされた通りに動作しているに過ぎず、エラーを検出しない。

論理エラーを修正する場合は①「何がバグであるか」を設計内容または要求事項とプログラムの動作結果との差異から判断する」ことになる。まずはバグの再現性を問うことから始め、どのような条件で顕在するのかを把握する。②「バグを再現させながら、設計内容および要求事項と合致するはずのプログラムを比較し、バグを特定」する。ただし、プログラムを動作させるだけではバグの特定が困難な場合はプログラムの一部を原因調査用の処理に変更することもある。なお、コンピュータの様子を人が直接確認することができないため、ハードウェア（センサ、アクチュエータ）、特にユーザーインターフェイスを頼りに情報を得ることになる（表1 参照）。差異を確認後は基本的に③「設計内容や要求事項に従い、プログラムを修正」を行うが、実現不可能だったと判断できる場合は③「設計内容もしくは要求事項の変更」を行うこともある。

表1 デバッグ手法の例

デバッグ手法名	利用方法	出来ること（○：可能，－：不可）		
		通過確認 ^{※1}	変数参照 ^{※2}	実行停止 ^{※3}
print 文	プログラムの節目に実行している処理内容が分かる文字列を表示する print 文を挿入する。	○	○	－
LED やブザー	プログラムの節目に「出力ポートの On-Off 制御（制御により LED が点灯，またはブザーが鳴動する）」を挿入する。	○	－	－
デバッガ	開発環境で用意されているデバッガを利用し，プログラムを実行する。	○	○	○
ログ保存	プログラムの節目に実行している処理内容が分かるログ保存用の処理を挿入する。	○	○	－

- ※1：コンピュータが処理した箇所が分かる。実行してほしいプログラムが意図したタイミングでコンピュータに実行されているかが判断できる。
- ※2：コンピュータが参照できる変数の値を確認できる。変数は条件分岐で参照されるため、どのようなタイミングで条件を満たすのか把握できるようになる。
- ※3：任意のタイミングでコンピュータがプログラムを処理することを停止することができる。停止中に変数の値を参照できる他、値の変更もできるため、強制的に条件分岐を満たすことで任意のプログラミングを実行しやすくなる。

3. 検定教科書に登場するデバッグ（修正）手法の調査結果

2024年8月時点で使用されている中学校技術科の教科書からデバッグについて、どのように取り上げているかを調査した。調査方法としては教科書の索引より“デバッグ”を探し、該当箇所を調べた。加えて、プログラミングに関するページを目次より探し、デバッグに関する記載内容を調べた。調査した結果、デバッグの定義の説明文“バグを修正する作業をデバッグといい、プログラムを作成するうえでは重要な作業になる”⁹⁾はあっても、具体的なデバッグ手法について紹介がないことが分かった（表2参照）。中学校技術科の教科書のみでデバッグ手法の紹介がない可能性があり、プログラミングを扱っている高等学校の情報科および工業科についても調査したが、ほとんどの教科書でデバッグ手法について紹介されていないことが分かった（表3参照）。

そもそも、中学校技術科の教科書ではデバッグ手法も含め、“修正方法を紹介しないことが普通なのではないか”¹⁰⁾¹¹⁾と思ひ、他の内容（内容A、BおよびC）においても「間違いを修正する方法」の紹介がないかを調査した。調査の結果、教科書によって異なるが、他の内容では「間違いを修正する方法」もしくは「修正が必要と判断するための基準」などが紹介されていた（表4参照）。よって、プログラミング教育のみ「間違いを修正する方法」の紹介がないことが分かった。

表2 技術科（中学校）の教科書に登場する「デバッグ手法」に関する記載内容

教科	出版社	教番	ページ	デバッグ手法の記載内容
技術・家庭科 (技術分野)	東京書籍	技術 701 ¹¹⁾	p.226	※なし。言葉の説明のみ。 ⁹⁾
	教育図書	技術 702 ¹²⁾	p.235	※なし。言葉の説明のみ。
	教育図書	技術 703 ¹¹⁾	—	※記載なし。
	開隆堂	技術 704 ¹²⁾	p.211	※なし。言葉の説明のみ。

表 3 情報科および工業科（高等学校）の教科書に登場する「デバッグ手法」に関する記載内容

教科	出版社	教番	ページ	デバッグ手法の記載内容
情報Ⅰ (共通教科情報科)	東京書籍	情Ⅰ 701 ¹³⁾	—	※記載なし。
	東京書籍	情Ⅰ 702 ¹⁴⁾	p.81	※概要の説明のみ。
	実教出版	情Ⅰ 703 ¹⁵⁾	p.133	※なし。エラーの定義あり。
	実教出版	情Ⅰ 704 ¹⁶⁾	p.133	※なし。エラーの定義あり。
	実教出版	情Ⅰ 705 ¹⁷⁾	—	※記載なし。
	実教出版	情Ⅰ 706 ¹⁸⁾	—	※記載なし。
	開隆堂	情Ⅰ 707 ¹⁹⁾	—	※記載なし。
	数研出版	情Ⅰ 708 ²⁰⁾	—	※記載なし。
	数研出版	情Ⅰ 709 ²¹⁾	—	※記載なし。
	日本文教出版	情Ⅰ 710 ²²⁾	p.163	※なし。構文エラーに関する記述あり。
	日本文教出版	情Ⅰ 711 ²³⁾ 情Ⅰ 712 ²⁴⁾	—	※記載なし。
	第一学習社	情Ⅰ 713 ²⁵⁾	—	※記載なし。
情報Ⅱ (共通教科情報科)	東京書籍	情Ⅱ 701 ²⁶⁾	—	※記載なし。
	実教出版	情Ⅱ 702 ²⁷⁾	p.75	状況を調べる JupyterLab のマジックコマンドが紹介されている。
	日本文教出版	情Ⅱ 703 ²⁸⁾	p.119	デバッグ時には処理順番および変数の値の確認が重要と述べている。また、デバッガが有する機能を紹介している。
専門教科情報科	実教出版	情報 701 ²⁹⁾	—	※記載なし。
	実教出版	情報 704 ³⁰⁾	—	※プログラムに関連する記載なし。
工業科	実教出版	工業 701 ³¹⁾	p.267	※概要の説明のみ。
	実教出版	工業 718 ³²⁾	p.65	※記載なし。修正を含めた作成チャートがある。
	実教出版	工業 719 ³³⁾	—	※記載なし。
	オーム	工業 723 ³⁴⁾	p.173	フローチャートをトレースすることを紹介している。
	オーム	工業 743 ³⁵⁾	—	※プログラムに関連する記載なし。
	実教出版	工業 766 ³⁶⁾	p.175	print 文によるデバグライトという名称が登場する。
	実教出版	工業 767 ³⁷⁾	p.230	※記載なし。言葉の説明とチャートが登場する。

表 4 中学校技術科の検定教科書に登場する「間違いの修正方法」に関する記載内容

教科	出版社	教番	ページ	修正方法の記載内容
A：材料と加工	東京書籍	技術 701 ⁹⁾	p.55	接着剤を使って修正する方法を紹介している。
			p.59	ベルトサンダや研磨紙で修正する方法を紹介している。
			p.60	接着剤を使って修正する方法を紹介している。
	教育図書	技術 702 ¹⁰⁾	—	※記載なし。
		技術 703 ¹¹⁾	p.10	誤った釘を抜く方法を紹介している。
	開隆堂	技術 704 ¹²⁾	p.71, p.74	木工用接着剤を用いた修正方法を紹介している。
			p.76, p.85	検査方法と修正方法を紹介している。
B：生物育成	東京書籍	技術 701 ⁹⁾	—	※記載なし。
	教育図書	技術 702 ¹⁰⁾	—	※記載なし。
		技術 703 ¹¹⁾	p.17	作物の育場面ごとの対処方法を早見表にして紹介している。
	開隆堂	技術 704 ¹²⁾	p.105	養分不足の症状が紹介されている。
			p.106	病害虫が発生した場合の対処方法が紹介されている。
			p.123	ミニトマトの栽培で病害虫が発生した場合の対処方法が紹介されている。
C：エネルギー変換	東京書籍	技術 701 ⁹⁾	p.151	回路計を用いた検査方法が紹介されている。
	教育図書	技術 702 ¹⁰⁾	—	※記載なし。
		技術 703 ¹¹⁾	p.25	はんだ吸い取り線で修正することを紹介している。
			pp.26-27	回路計の操作方法を紹介している。
	開隆堂	技術 704 ¹²⁾	p.165	電気機器が動作しないときに回路計で検査する方法が紹介されている。
D：情報	※「表 2」を参照。			

4. おわりに

本調査ではプログラミング教育の課題である難しさを感じてしまう対策としてデバッグ手法を身に付けることが有効と仮説を立て、「検定教科書にデバッグ手法が紹介されているか」を調べた。その結果、検定教科書にはデバッグ手法の紹介がないことを確認した。これによりデバッグ能力が養われず、プログラミング教育に難しさを感じる要因になっている可能性が示唆される。さらに、このデバッグという「問題となった原因を特定し、修正する」手段が教科書から学べず、また教員も指導できないのであれば、プログラミング教育のねらいである“問題を見いだして課題を設定し解決する力の育成”^{2) 38)}についても実現できていないのではないかと危惧する。

今日のプログラミング教育には課題があるが、世に存在するプログラミングによって生み出された、多くの製品やサービスは安定して稼働している。また、ソフトウェア開発においてデバッグは当然のように行われているため、デバッグ手法のノウハウはあるはずである。そのノウハウを活かすことが出来れば、プログラミング教育の改善は出来るはずである。そのため、今後の課題としては授業実践を通じて、学校のプログラミング教育に適用できるデバッグ手法を考案していく。

参考文献

- 1) 文部科学省, “【技術・家庭編】中学校学習指導要領 (平成 29 年告示) 解説,” 2017, p.23.
- 2) 文部科学省, “【技術・家庭編】中学校学習指導要領 (平成 29 年告示) 解説,” 2017, p.152.
- 3) 特定非営利活動法人みんなのコード, “全国の学校教育における「プログラミング教育実態調査」を公開ー小・中学校教員 2,400 名、子ども・保護者 3,000 組を対象に意識調査を実施ー,” 2 12 2021. オンライン). Available: <https://code.or.jp/news/20211202/>. アクセス日: 11 2024).
- 4) 特定非営利活動法人みんなのコード, “プログラミング教育 実態調査報告書,” pp.33-36, 2021.
- 5) 楠見孝・西川一二・齊藤貴浩・栗山直子, “プログラミング教育の授業実践に対する小中学校教員の期待と意欲,” 日本教育工学会論文誌, 第 44 巻, 第 2, pp.265-275, 2020.
- 6) 小泉匡弘・藤川聡・渡壁誠, “自律型ロボット教材とした「プログラムによる計測・制御」に関する学習の教育的効果及び難しさの要因,” 日本教材学会, 第 28 巻, pp.59-68, 2017.
- 7) 株式会社インセプト, “IT 用語辞典 e-Words,” 31 8 2024. オンライン). Available: <https://e-words.jp/w/コンパイラコンパイラ.html>. アクセス日: 1 1 2025).
- 8) 宇都宮魁斗・秋山楽登・近藤将成・亀井靖高・鶴林尚靖, “初学者のためのデバッグ作業支援に向けた大規模言語モデルの初期評価,” ソフトウェア工学の基礎ワークショップ論文集, pp.63-71, 2023.
- 9) Scratch 財団, “Scratch,” オンライン). Available: <https://scratch.mit.edu/>. アクセス日: 1 11 2024).
- 10) 田口浩継ほか 80 名, “4 コンテンツのプログラムを制作しよう,” 著: 新しい技術・家庭 [技術分野] 未来を創る Technology, 東京書籍, 2020, p.226.
- 11) 田口浩継ほか 80 名, 新しい技術・家庭 [技術分野] 未来を創る Technology, 東京書籍, 2020.
- 12) 中村祐治ほか 47 名, New 技術・家庭 家庭分野 くらしを創造する, 教育図書, 2020.
- 13) 中村祐治ほか 47 名, New 技術・家庭 家庭分野 くらしを創造する 技能ハンドブック, 教育図書, 2020.
- 14) 竹野英敏ほか 118 名, 技術・家庭[技術分野], 開隆堂, 2020.
- 15) 赤堀侃司・東原義訓・坂元章ほか 28 名, 新編情報 I, 東京書籍, 2022.
- 16) 赤堀侃司・東原義訓・坂元章ほか 29 名, 情報 I Step Forward!, 東京書籍, 2022.
- 17) 萩谷昌也ほか 10 名, 高校情報 I Python, 実教出版, 2022.
- 18) 萩谷昌也ほか 10 名, 高校情報 I JavaScript, 実教出版, 2022.
- 19) 萩谷昌也ほか 10 名, 最新情報 I, 実教出版, 2022.
- 20) 萩谷昌也ほか 8 名, 図説情報 I, 実教出版, 2022.
- 21) 本郷健・松原伸ほか 15 名, 実践 情報 I, 開隆堂, 2022.
- 22) 坂村健ほか 12 名, 高等学校 情報 I, 数研出版, 2022.
- 23) 坂村健ほか 12 名, 情報 I Next, 数研出版, 2022.
- 24) 黒上晴夫・堀田龍也・村井純ほか 59 名, 情報 I, 日本文教出版, 2022.
- 25) 黒上晴夫・堀田龍也・村井純ほか 54 名, 情報 I 図解と実習ー図解編, 日本文教出版, 2022.
- 26) 黒上晴夫・堀田龍也・村井純ほか 54 名, 情報 I 図解と実習ー実習編, 日本文教出版, 2022.
- 27) 山口和記ほか 13 名, 高校教科書 高等学校 情報 I, 第一学習社, 2022.
- 28) 赤堀侃司・東原義訓・坂元章ほか 28 名, 情報 II, 東京書籍, 2022.
- 29) 黒上晴夫・堀田龍也・村井純ほか 39 名, 情報 II, 実教出版, 2022.
- 30) 萩谷昌也ほか 4 名, 情報 II 学習ノート新課程版, 日本文教出版, 2022.

- 31) 伏見正則ほか7名, 情報産業と社会, 実教出版, 2022.
- 32) 国廣昇ほか6名, 情報セキュリティ, 実教出版, 2022.
- 33) 山下省蔵・内藤善文・扇柳政則ほか19名, 工業技術基礎, 実教出版, 2020.
- 34) 堀桂太郎ほか9名, 工業情報数理, 実教出版, 2021.
- 35) 堀桂太郎・美馬誠ほか7名, 精選工業情報数理, 実教出版, 2021.
- 36) 石塚満ほか4名, 工業情報数理, オーム, 2021.
- 37) 横山明彦ほか2名, オーム, 電力技術 2, 2022.
- 38) 浅井英史・岩崎浩文ほか5名, ソフトウェア技術, 実教出版, 2023.
- 39) 浅井英史・岩崎浩文ほか5名, コンピュータシステム技術, 実教出版, 2023.
- 40) 文部科学省, “【技術・家庭編】中学校学習指導要領(平成29年告示)解説,” 2017, pp.52-58.
- 41) 松尾春紀・池田翔・亀井靖高・佐藤亮介・島田敬士・鶴林尚靖, “教育支援の適用に向けた自動バグ修正手法の性能調査,” コンピュータ ソフトウェア, 第 38 巻, 第 4, pp.16-22, 2021.
- 42) 秋山楽登・中村司・近藤将成・亀井靖高・鶴林尚靖, “プログラミング初学者のバグ修正履歴を用いたデバッグ問題自動生成の事例研究,” ソフトウェア工学の基礎ワークショップ論文集, pp.13-22, 2021.
- 43) 寺田実, “デバッグのためのベンチマークプログラム,” 第 53 回プログラミング・シンポジウム予稿集, pp.107-114, 2012.
- 44) 山本頼弥・野口靖浩・小暮悟・山下浩一・小西達裕・伊東幸宏, “場当たりのなデバッグを行ってしまう学習者に体系的デバッグ手順を指導する授業パッケージと学習支援システムの構築,” 教育システム情報学会誌, 第 35 巻, 第 1, pp.21-37, 2018.
- 45) 三輪理人・梅田恭子, “デバッグに着目した小学校プログラミングの実践,” 愛知教育大学研究報告. 教育科学編, 第 3 巻, pp.109-112, 2024.

— Abstract —

Investigation of Debugging Methods Mentioned in Programming Education Textbooks for Junior High School Technology Classes in Japan

KENICHI Iwaki (1) FUJIMOTO Koji (2) NOGUCHI Satoshi (3)

Programming education is offered in junior high school technology classes. However, survey results show that some teachers and students find programming education difficult. The reason for this is thought to be that "they don't know how to program to achieve the desired results." Thinking that debugging techniques are necessary to address this issue, I looked into textbooks.

However, there were no junior high school technology textbooks that introduced debugging techniques. Furthermore, I also looked into information and industrial textbooks that deal with programming in high schools, but most textbooks did not introduce debugging techniques. Furthermore, I thought it was normal for junior high school technology textbooks to not introduce "methods to correct mistakes" such as debugging, but other content introduces "correction methods" and "criteria for determining whether corrections are necessary."

From the above, it is suggested that one of the reasons why programming education is difficult is that "debugging techniques cannot be learned from textbooks."