

プログラムを書かせないプログラミング学習支援の可能性検証 —命令を組み立てることによるプログラム作成方式—

Examining a Possibility of Programming Learning Support Without Writing a Program

- A Method to create a program by constructing the parts of program instructions -

石井 元規^{*1}, 岩井 健吾^{*1}, 松本 慎平^{*1}, 平嶋 宗^{*2}, 林 雄介^{*2}

Motoki ISHII^{*1}, Kengo IWAI, Shimpei MATSUMOTO, Tsukasa HIRASHIMA, Yusuke HAYASHI^{*2}

^{*1}広島工業大学 情報学部

^{*11}Faculty of Applied Information Science, Hiroshima Institute of Technology

Email: b213011@cc.it-hiroshima.ac.jp

^{*2}広島大学

^{*2} Hiroshima University

あらまし：本研究では、学習者は予め与えられているカード群から、必要なものを選択し、適切な順序でカードを並び替えることによってプログラミング演習を行うことができるシステムを開発した。開発方法は、Webブラウザからの利用に対応するため、主にRubyを使用した。開発したシステムにより、既存言語にも対応した形であっても、プログラミング初心者の学習を妨げる原因の一つであるコーディングの負荷を取り除くことができた。

キーワード：コーディング負荷、カード演習、プログラミング、学習支援システム

1. はじめに

世界中においてコンピュータサイエンス専攻の入学の内、30~60%はプログラミングができずに落第しているという現状がある⁽¹⁾。これは、今後さらにプログラミングの重要性が高まることを考慮すると解決すべき重要な問題であるといえる。この問題を解決するためには、プログラミング学習を妨げる原因の特定を行い、その原因を取り除いた学習支援を行うべきであると考えられる。

本研究の目的は、タイプミスに起因する文法エラーを取り除くプログラミング演習システムの設計及び開発を行うことで、プログラムを書かせないプログラミング学習支援の可能性を検証することである。具体的には、タイプミスに起因する文法エラーを取り除くために、モンサクン方式に基づいた直観的なカード操作による演習によって、プログラムを書く必要のないプログラミング演習の提案を行った。さらに、実際に大学生に対してシステムの運用を行い、システムに関するアンケートを実施した。

2. 背景及び関連研究

2.1 プログラミング学習の妨げとなる要因

先行研究の中において、プログラミングの初学者は、学習の初期段階でつまづくことが多く、とりわけ学習の初期段階では、プログラムのタイプミスに起因する文法エラーが多発することが指摘されている⁽²⁾。これから、プログラミング学習を妨げる原因の一つは、タイプミスに起因する文法エラーであると考えられる。そのため、プログラミング初学者が円滑な学習を行うには、タイプミスに起因する文法エラーを取り除く必要性があると考えられる。

2.2 関連研究

タイプミスに起因する文法エラーを取り除いている関連研究としてScratchのようなビジュアルプログラミング言語が存在する。Scratchでは、図形等を用いてプログラムを組み立てるので、マウス等を利用し、簡単にプログラムを組むことが可能となっている。そのため、コーディング作業を行う必要がなく、タイプミスに起因する文法エラーは取り除かれている。コーディング負荷を取り除いたことで、プログラミング初学者にとって学習のハードルを下げた点でScratchのような学習方法は有用であると考えられる。しかしながら、Scratchには、大規模なソースコードを書くことが難しいという問題点も存在する。そのため、どれだけScratchを学習しても、ある一定の規模のソースコードを書く場合には、他の言語を用いる必要が生じる。その際に、新しい言語を学ぶプログラミング初学者は、再度タイプミスに起因する文法エラーに直面することが考えられる。これを回避するためには、既存の言語であってもScratchのように簡単にプログラムを組み立てることができるような支援を行う必要があると考えられる。本研究では、Scratchのような独自言語ではなく、C言語のような既存言語への対応を試みた。

カード操作による学習支援を実現している関連研究として作問学習支援システム「モンサクン」がある⁽³⁾。モンサクンでは、問題を解くのではなく、作ることによって演習を行う作問学習を行うことが可能となっている。そして、カードを並び替えるという簡単な操作だけで作問を行うことができる直観的なインターフェース設計が実装されている。本研究では、このようなカード操作による演習方式をモンサクン

<p>1-3 変数名の規則 (使用可能な記号)</p> <p>問題 処理結果で1つ目の変数の値が5,2つ目の変数の値が50となるようにカードを並び替えて下さい。</p> <pre>#include<stdio.h> int main(void){ //この中の処理を並び替えて下さい return 0; }</pre> <p>正解</p> <pre>int no, math_test; no = 5; math_test = no * 10;</pre> <p>ダミー</p> <pre>int No, math-test; No = 5; math-test = No * 10;</pre>	<p>2-3 整数同士の演算</p> <p>問題 処理結果がa = 12, b = 0, c = 6となるようカードを並び替えて下さい。</p> <pre>#include<stdio.h> int main(void){ int a, b, c; //この中の処理を並び替えて下さい return 0; }</pre> <p>正解</p> <pre>c = 6; a = c + 6; b = c / a;</pre> <p>ダミー</p> <pre>b = a / c; a = b / a - 1; b = a / c - 1;</pre>	<p>3-3 剰余演算子%</p> <p>問題 処理結果がa = 3, b = 3, c = 11となるようカードを並び替えて下さい。</p> <pre>#include<stdio.h> int main(void){ int a, b, c; //この中の処理を並び替えて下さい return 0; }</pre> <p>正解</p> <pre>c = 11; a = c / 3; b = a % 5;</pre> <p>ダミー</p> <pre>a = c % 3; a = a / 5; b = a / c;</pre>	<p>4-3 初期化の理解</p> <p>問題 処理結果がa = 45976, b = 4, c = 24035となるようカードを並び替えて下さい。</p> <pre>#include<stdio.h> int main(void){ int a, b, c; //この中の処理を並び替えて下さい return 0; }</pre> <p>正解</p> <pre>c = ((777 - 395) / 4) * (888 - 635); a = (c - 7382) + (c - 18746) + (c - 1); b = ((a + a) / c) + ((c + c) / a);</pre> <p>ダミー</p> <pre>c = (200 - 180) + (20 - 18) + (c + 1); a = (a - 5382) + (a - 20746) + (a - 1); b = ((c + c) / b) + ((c + c) / a);</pre>	<p>5-3 インクリメント</p> <p>問題 処理結果がa = 7, b = 7, c = 6となるようカードを並び替えて下さい。</p> <pre>#include<stdio.h> int main(void){ int a, b, c; //この中の処理を並び替えて下さい return 0; }</pre> <p>正解</p> <pre>c = 5; a = ++c; b = ++a;</pre> <p>ダミー</p> <pre>c = c++; c = b++; b = c++;</pre>
--	---	---	---	--

図2 出題された問題の一例 (各分野からそれぞれ1問ずつを示している)

方式と呼ぶことにする。また、本研究では、このモンサクン方式に基づいてカード操作によるプログラミング演習システムの設計及び開発を試みた。

3. プログラミング学習支援システム

プログラミング学習支援システムでは、カードを並び替えることによってプログラムを組み立てる。現在のシステムが出題する問題は、以下のようなC言語に関する基本的な文法に関するものである。本提案のプログラミング学習支援システムの特徴をまとめると大きく三つある。一つ目は、コーディングを行う必要がないという点である。二つ目は、フィードバックを返すことが可能という点である。三つ目は、履歴データを取得することが可能という点である。図1は、本提案におけるプログラミング学習支援システムのインターフェースである。

実験は、C言語のプログラミングの授業を受講している広島工業大学情報学部の1年生103名を対象にして行った。演習時間は25分とした。また、問題の再挑戦と問題のパスは可能である。アンケートは演習を行った直後に行った。基礎的な文法の問題である、変数名、整数型、剰余、初期化、インクリメントの5分野から出題した。出題した問題の一例を図2に示す。なお、各分野3問ずつ出題したので、合計15問の出題を行った。

アンケート形式は、六段階リカード尺度を採用した。また、アンケート内容としては、プログラミングカード演習が学習しやすいか、コーディング演習と比較してどうか、どのような学習効果が得られると思うか等の観点から16項目をアンケートで調査した。アンケート結果の一部を図3に示す。図3は、プログラミングカード演習の学習しやすさに関するアンケート結果である。このことから、組み立てることによるプログラム作成方式の有用性が示唆されたと考えられる。

4. おわりに

本研究では、プログラムを書かせないプログラミング学習支援の可能性を検証した。

謝辞

本研究は、独立行政法人日本学術振興会科学研究

費助成事業(若手(B) 13304922, 基盤研究(C) 26350296)の助成を受けて実施した成果の一部である。ここに記して謝意を表します。

参考文献

- (1) Saeed Dehnadi, Richard Bornat: "The camel has two humps", Middlesex University Working Paper(2006).
- (2) 岡本雅子, その他: "プログラミングの写経型学習過程を対象としたつまずきの分析とテキスト教材の改善, 京都大学高等教育研究第19号, pp.47-57 (2013).
- (3) 平嶋宗: "作問学習のモデル化", The 23rd Annual Conference of the Japanese Society for Artificial Intelligence, pp.1-3 (2009).



図1 システムのインターフェース

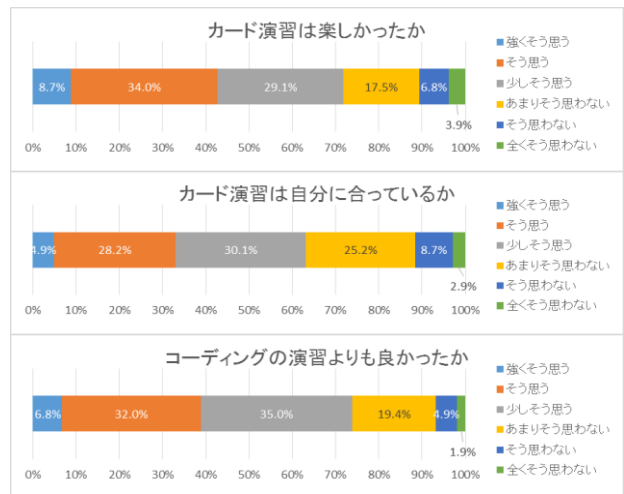


図3 アンケート結果一例