

# プログラミング演習におけるブロック型インタフェースの導入と実践

三重野 剛<sup>\*</sup>, 三浦 元喜<sup>\*\*</sup>

<sup>\*</sup> 九州工業大学 工学部 総合システム工学科

<sup>\*\*</sup> 九州工業大学 大学院工学研究院 基礎科学研究系

## Introducing Block Interface for Programming Exercises

Tsuyoshi MIENO<sup>\*</sup>, Motoki MIURA<sup>\*\*</sup>

<sup>\*</sup> Department of Integrated System Engineering, School of Engineering, Kyushu Institute of Technology

<sup>\*\*</sup> Department of Basic Sciences, Faculty of Engineering, Kyushu Institute of Technology

**あらまし** - プログラミング初学者は、タイピングの遅れやエラーの発生によって、授業についていけない状況が発生する。大学2年生を対象としたプログラミング演習において、ブロック型とテキスト型の2種類のインタフェースを比較する実験を行った。インタフェースの習熟度や、作成するプログラムの複雑さがスコアに影響する可能性が示された。

**キーワード** - プログラミング, ブロック型インタフェース, エディタ環境

### 1 はじめに

プログラミングの学習においては、学習者の経験やスキルの差が、理解度に大きく影響する。タイピングの遅れによって説明を聞き逃したり、タイプミスから生じるエラーの復旧に時間がかかったりすることで、理解が追いつかなくなることが原因と考えられる。ひとつの解決法として、正確かつ高速なタイピングの訓練をすることが挙げられる。しかし、訓練には時間を要するため、タイピングスキルの個人差による学習への影響は依然として大きい。

初学者を対象としたプログラミング環境には、ブロック型インタフェースが採用されることが多い。ブロックの配置により、プログラムの構造をテキスト記述に比べて、視覚的に理解しやすい。またマウスによる直接的な編集操作が可能であるという特徴がある。ブロック型インタフェースを用いることで、テキスト型の環境に比べて、学習や演習の効率を高めることができる可能性がある。

本研究では、ブロック型インタフェースを導入した Fortran プログラミング学習システム Scratchly を開発し、その効果を検証した。

### 2 ブロック型インタフェース

Scratchly は、Google Blockly<sup>(1)</sup>を利用した、ブロック型のプログラミングインタフェースである。ソースコードの出力として、Fortran 言語に対応させる

ため、変数宣言や、implicit none 宣言、キーボードからの入力と変数への割り当てに対応するブロックを追加している。画面構成を図1に示す。学習者は、画面左側の灰色のエリアでクリックすると、該当するブロックの一覧が表示される。学習者がブロックを選択し、ドラッグすると、プログラム編集領域に配置することができる。通常のブロック型インタフェースと同様、ブロック同士のスナッピング機能によって、ブロックの配置や接続の手間を軽減している。ブロックの配置を行うと、画面右のソースコードが自動的に更新される。画面右下のテキストエリアに入力値を設定し、「実行」ボタンを押すと、プログラムの動作を確認できる。また学習者は、「サーバでチェック」ボタンを押すことによって、プログラムに規定の値を入力して、動作が正しいかどうかを確認することができる。

### 3 実験

大学2年生を対象とした、実験授業を行った。2問の練習問題(Q1,5)と、6問の演習問題(Q2~4, 6~8)と、2問のテスト問題(Q9~10)を準備し、90分の講義時間で制限時間内に回答してもらった。演習問題の前半は、条件分岐に関する問題で、後半は繰り返しを利用して回答する問題とし、難易度は問題番号が増えるに従って高くなるようにした。ブロック群は、図1に示すシステムを利用し、テキスト群は、従来のテキストエディタ(Webブラウザ上で動作す

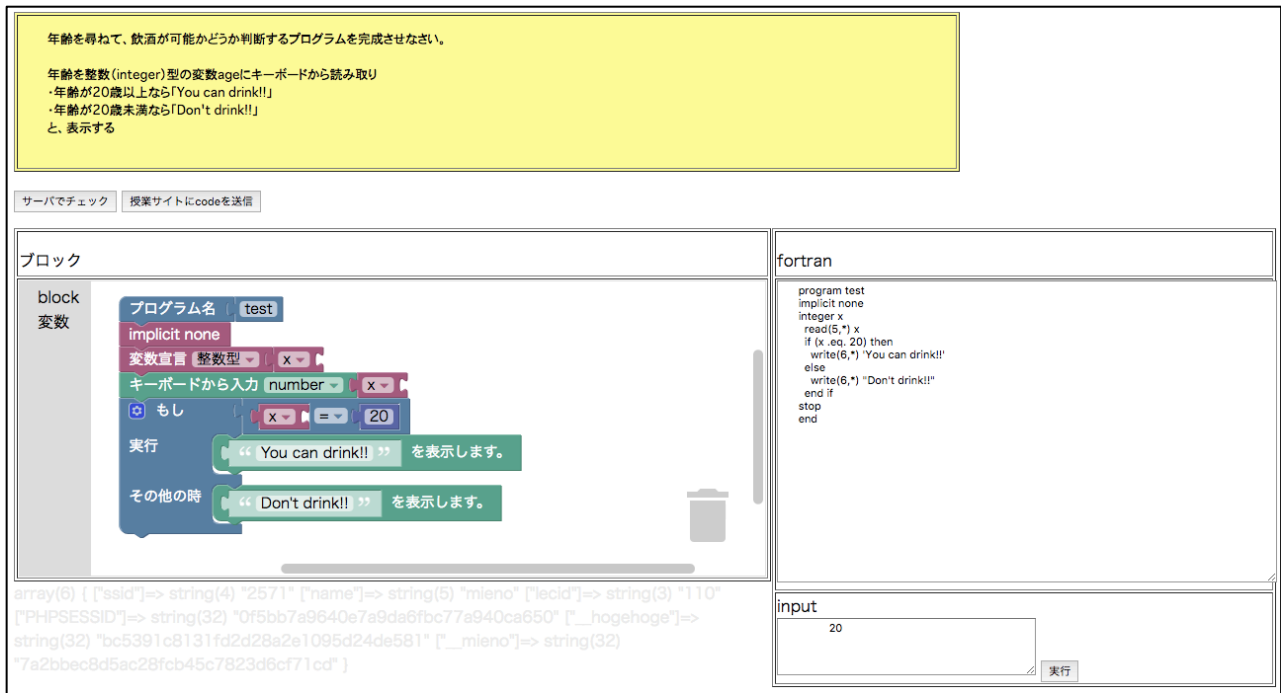


図 1：ブロック型インターフェースによる、Fortran コードの出力

るエディタ、または gedit) を利用した。両群とも、回答は正解するまで何度でも修正可能であった。

#### 4 実験結果と考察

提出された各回答について、4点満点で採点した。練習問題を除く、問題ごとの平均点と、標準誤差を図2に示す。横軸は問題番号で、縦軸はスコアを示す。問題番号のうしろの\*は、Welch の t 検定による 5% 有意を示している。なお、受講学生の数および、提出された回答数 (表 1) は問題・条件によって異なる。Q7 は問題のチェック機構に不備があったため、分析から除外した。

ブロック群は、当初ブロック操作のインターフェースに十分慣れていなかったため、複数の else 句が要求される問題 (Q3) や、複雑な構造が要求される問題 (Q10) の正解率が低かったと考えられる。Q4 や Q8 の段階では、操作の習熟がある程度進んだことによ

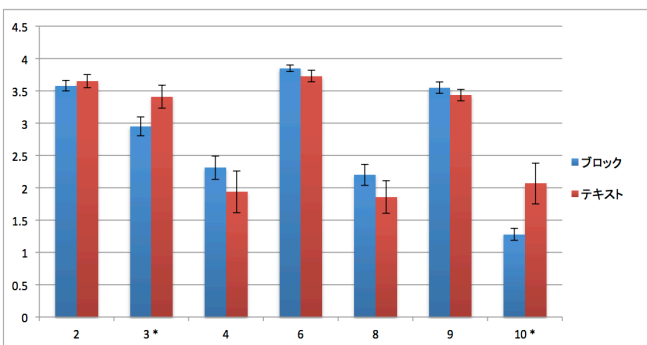


図 2：スコアの平均と標準誤差

って、定型的なプログラムを短時間で正確に作成できるようになったことが、スコアの上昇に寄与したと考えられる。

表 1：学生数と、提出された回答数

	人数	Q 2	Q 3	Q 4	Q 6	Q 8	Q 9	Q 10
ブロック	139	117	84	60	134	65	72	49
テキスト	79	72	46	16	75	21	76	15

#### 5 まとめ

プログラミング演習を通じて、ブロック型インターフェースの効果を検証した。今回の実験ではインターフェース毎の習熟度の違いに加え、短時間に多数の問題を解かせた影響によって、本来のインターフェースがもつ特性を十分検証できなかったと言え難い。

ICT 活用教育を組織的に導入・展開するうえでは、利用者の習熟度や学習段階を考慮しつつ、インターフェースを継続的に改善していくことが重要であると考えている。引き続きブロック型インターフェースの有効性について検証していきたい。

#### 参考文献

- (1) Neil Fraser. Ten Things We've Learned from Blockly. In 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), pp. 49–50, Oct 2015.