R-4

制御文の実行順序を可視化するプログラム理解支援ツール

橋浦研究室　　1165140 大沼 克海　　1165440 弓田 快

1. 研究背景

プログラミング初学者の中には，文法を曖昧にしか理解できていない，教科書にかいてあるプログラムをそのままの形で覚えているなど，正しく理解できていない学生がいる．今泉ら[1]の研究でも述べられているように，一般的にはプログラミング言語の文法の学習，基本的なアルゴリズムとその実装方法の学習，アプリケーション開発といった順序でプログラミング教育は行われる．そのため，文法を学習した時点で正しい理解ができていないと以後の内容に対して理解することができなくなってしまう．よって，プログラミング初学者は文法に対する理解を深めることが重要である．このことから，文法の理解を支援するために実行順序を可視化するツールを開発しようと考えた．

2. 研究目的

本研究は，可視化ツールを用いて制御文の実行順序の理解を支援するためにソースコードに実行順序を表示する．この支援方法によって，実行順序の理解を支援することができる．

3. 提案手法

3.1. 学習の流れ

プログラミング学習の際に，制御構文を学ぶときがある．その時にツールにより実行順序を表示することによって，コードに対しての実行順序が分かり理解の支援につなげる．

3.2. 上手になるために必要なこと

長谷川ら[2]の研究で，ソースコードの動作イメージができる能力はプログラム学習の達成度と関係があると述べられている．そのため，ツールを用いて上達する部分はプログラムを読みイメージする能力と制御文に対する理解である．制御文に対する理解は，初学者が if 文などを初めて触れる時に試しにソースコードを使用することでソースコードがどのように実行されるかの理解を深めることができる．このツールは実行順序を読むことによって制御文などの理解を深めることを想定しているため，初学者がツールで表示された実行順序をソースコードと共に読む必要がある．

4. 実現方法

本ツールには前述のプログラミング学習の際に実行順序をオーバーレイ表示する（図 1）ツールを開発した．本ツールは UI 要素を追加する機能などが開発でき，リリースまでを行うことができる Visual Studio の拡張機能として開発を行ったため，Visual Studio で動作するツールとなっている．

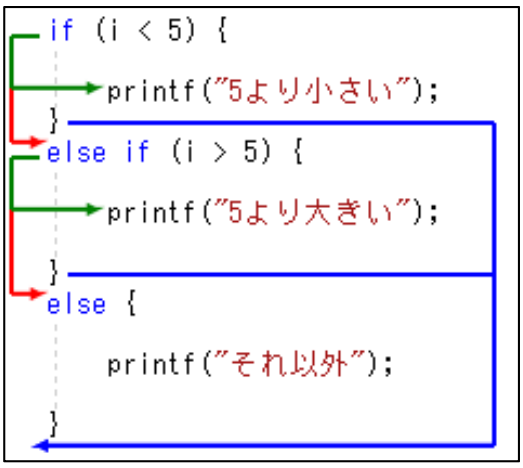

図 1．矢印の描画

5. 実験

本研究で作成したツールの有用性を検証するとともに，本ツールを用いた学習方法である自身でコードを書いて実行順序を見ることと，描画された実行順序を説明されることに差があるのか検証するために日本工業大学情報工学科で行われている「プログラミングⅡ」の受講生 22 人を対象とし実験を行った．

その実験の手順を以下に示す．

1. 実験の説明（15 分）
2. ツール使用前テスト（20 分）
3. ツールを使用し学習（45 分）
4. ツール使用後テスト（20 分）

ツール使用前後で出すテストで出題する問題の範囲は本ツールが対応している if 文と for 文とする．具体的には if 文，for 文，for 文の入れ子構造とし，実行順序について問う問題を出題する．問題の具体例は図 3 に示した．問題の回答方法は，四角で囲まれた処理の順序に対し矢印を用いて書いてもらう．

「ツールを使用しての学習」に関しては，本ツールを用いて出題された問題に対する実行順序を学習するものであり，学習時には自分と共にコードを入力しながらツールの説明を行うものとなっている．

```
#include <stdio.h>
int main(void){
    int i ;
    scanf("%d",&i);

    if(i < 5){

        printf("i は 5 より小さい");
    }
    else if(i > 5){

        printf("i は 5 より大きい");
    }
    else{

        printf("どちらでもない");
    }
    return 0;
}
```

図 2. 問 1 の問題例

6. 評価方法

評価は，ペーパーテストでの正解数によって行う．検定による有意水準は 0.05 とする．

7. 結果と考察

本ツールが全体の平均として影響がでるか，描画あり学習と説明のみ学習 2 つのテスト結果に対して F 検定を行った．帰無仮説H_0を「ツール学習による正解数の分散の差はない」とし，対立仮説H_1を「ツール学習による正解数の分散に差がある」として検定を行った．

表 1. 学習後テストによる F 検定の結果

	描画あり	説明のみ
平均	6.875	5
分散	36.125	14.30769231
観測数	8	14
自由度	7	13
観測された分散比	2.524865591	
P(F<=f) 片側	0.070906057	
F 境界値 片側	2.832097502	

その結果，表 1 となり p 値が 0.05 より大きいため対立仮説H_1が棄却され帰無仮説H_0が採択される．よって，等分散を仮定した t 検定を行う．帰無仮説H_0を「ツール学習による正解数の全体平均に差はない」とし対立仮説H_1を「ツール学習による正解数の全体平均に差はある」として検定を行った結果，表 2 より p 値が 0.05 より大きいため対立仮説H_1が棄却され帰無仮説H_0が採択される．よって，テストの正解数の全体平均に差はなく，ツール学習は全体への影響がないという結果となった．

表 2. 学習後テストによる T 検定の結果

	描画あり	説明のみ
平均	6.875	5
分散	36.125	14.30769231
観測数	8	14
プールされた分散	21.94375	
仮説平均との差異	0	
自由度	20	
t	0.903115965	
P(T<=t) 片側	0.18860681	
t 境界値 片側	1.724718243	
P(T<=t) 両側	0.37721362	
t 境界値 両側	2.085963447	

ツールによって学習に差が出るかを調べるため，テスト項目を 6 つに分けてχ^2検定を行った．また，データ数により Yates の補正をかける．

表 3.「内部処理から末尾処理」による集計表

内部処理から末尾処理	正解数	不正解数	合計
描画あり	5	27	32
説明のみ	1	55	56
合計	6	82	88

表 4.「内部処理から末尾処理」χ^2検定結果

x二乗値	6.138792
Yates	4.153746
p値	0.041543

表 3 のクロス集計表より，χ^2検定を行った．その結果，「内部処理から末尾処理」の部分に関して表 4 より p 値が 0.05 以下となりツール学習による差が出るということが分かった．

8. まとめと今後の課題

本研究のツールでは，一部ではツール学習により差がでることが分かったが，全体平均として影響がでないという結果となった．原因として，描画方法や説明に必要な機能が不足していることが考えられる．具体的には，現在サポートしていない部分の実行順序の表示や，ステップ実行の際に次の処理への実行順序を表示するなどよりコードの動きに対応した機能を追加していくことが課題となる．

文　献

[1] 今泉 俊幸, 橘浦 弘明, 松浦 佐江子, 古宮 誠一, “ブロック構造の可視化によるプログラミング学習支援環境 Azur ～関数の動作の可視化～,”情処研報ソフトウェア工学（SE）, Vol.2010-SE-167,No.11, pp.1-7, Mar.2010.

[2] 長谷川 聡, 山住 富也, 小池 慎一, “プログラミング教育における制御構造のイメージと理解度について,” 情報処理学会論文誌, Vol.39, No.4, pp.1180-1183, Apr.1998.