プログラミングのスキル階層に関する研究

山本 三雄†

本論文では，サンプルプログラムから修正・改造する演習のときに使われるスキルを「プログラム修正・改造スキル」として提案する．そして，「プログラム修正・改造スキル」と文献[1] の各プログラミング関連スキル"Basics","Data","Tracing1","Tracing2","Sequence","Explain","Writing"との相関を測定して，「プログラム修正・改造スキル」がスキル階層のどの位置にあるのかを明らかにする．これによってコードが書けない学習者に対して，どのスキルを補強すればよいかを示せるプログラミング学習法の知見を得ることを目的とした研究の研究デザインを示したものである．

Research on programming skill hierarchy

MITSUO YAMAMOTO†

In this paper, it proposes the skill used when maneuvering of the correction and remodeling from the sample program as "Program correction and remodeling skill". And, it is clarified at which position of the skill hierarchy "Program correction and remodeling skill" is measuring the correlation of the skill related to each programming of "Program correction and remodeling skill" and paper[1] "Basics", "Data", "Tracing1", "Tracing2", "Sequence", "Explain", and "Writing". As a result, it is the one that the study design of the research to aim to obtain the finding of the programming study method that can show which skill only has to be reinforced to the learner who cannot write the code was shown.

1. はじめに

高度情報化社会では，情報システムを構築できる高度な情報処理技術者が必要とされている．これにともない大学・情報系専門学校・情報系企業などでは，実用的なプログラミングスキルの習得を目指す教育が行われている．また，小中高等学校では情報リテラシーの習得に力を注がれ，より若年から興味や関心を持つ機会があることから，早期からプログラミングスキルの習得を始める学習者がいる．このようにプログラミングの習得に興味や関心を持って学びたいと思う多様な学習者に裾野が広がったことは，より学習者の適性や基礎的なスキルに合った効果的なプログラミング教育手法の必要性が増しているといえる．

1.1 プログラムが書けない学習者

初学者の中にはプログラムの文法や命令の記述方法，学んだサンプルプログラムやその類題をプログラミングする演習などは理解してできる人でも，見た目や問われ方の違う演習，学んだことの組み合わせ等が必要な応用的なプログラミング演習では，どうしてよいか分からないという学習者がいる．教育機関の実情や指導方針によっても異なる場合もあるが，求められる習得スキルは，サンプルプログラムとは全く異なる機能の応用的な機能を持つプログラムを，初めの状態からプログラミングする能力である．このような応用的なプログラムが書けない学習者がいることから，プログラムを理解する能力と応用的なプログラムを初めの状態からプログラミングする能力に隔たりがあるのではないかと感じられる．

1.2 プログラミング関連スキル階層

文献[1]では，プログラミング関連スキルは，用語や文法を理解している能力，プログラムの流れをトレースできる能力，与えられた部分的な組合せでプログラムを完成させる能力など，様々なスキルがあること．そして，それらのスキル間には相関があり，各スキル間は階層として考えられることが示されている．

1.3 本研究の目的

本研究の目的は，文献[1]で提案されていなかった「プログラム修正・改造スキル」を提案して，他のスキルとの相関について，次の点について明らかにする．

- 「プログラム修正・改造スキル」は，文献[1]のスキル階層の中でどの位置にあるか．

このことを明らかにすることで，学習者がどこでつ

*† 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo

まずいているのか（どのようなスキルを補強すればよいか）の把握に利用することができる．このような観点でプログラミング教育の向上に役立てたい．

1.4 研究方法

各スキルに対応する設問からなるプログラミングに関する筆記テストを初心者プログラミング学習者に実施する．スキル毎の設問ごとに成績の相関を統計的に算出することで，「プログラム修正・改造スキル」の文献[1]に中での階層を明らかにする．

1.5 本論文の構成

本論文の構成は以下のようになっている．第2章では，関連研究，プログラミングスキル階層，文献[1]で行った Java 問題について説明する．第3章では，「プログラム修正・改造スキル」の定義と測定問題について説明する．第4章では，実験計画，分析項目，実験結果の予想と仮説．第5章はまとめである．

2. 関連研究

本章では，先行研究と文献[1]で紹介されているプログラミング関連スキルに関して紹介する．

2.1 文献[1]で示されている先行研究

文献[1]で紹介されている先行研究について説明する．ITiCSE2004 会議で Leeds Group は，与えられた複雑さのコードを読める能力は，同等の複雑さのコード記述スキルであるとした．そしてプログラミングスキルの複数レベルの階層の存在の可能性を開いた．またコード記述スキルの下の階層に2つあり得ることを報告している．

BRACElet group は，Leeds Group の結果を基にプログラミング関連スキル階層化の中間レベルを徹底的に調査して，プログラムスキルには，複数レベルのスキル階層があること．コードを読むスキルとそれを説明するスキルの関係は，プログラム関連スキル階層の中間スキルであることを報告している．

Philpott,Robbins and Whalley(2007)は，コードトレーススキルはコード読解スキル以前のスキルであることを報告している．

Mike Lopez,Jacqueline Whalley,Raymond Lister(2008)は，コードトレーススキルとコード記述スキルの間に強い関連性が見られること，コード読解スキルとコード記述スキルの間に関連性が見られることを報告している．

2.2 プログラミング関連スキル階層とスキル測定問題

2.2.1 プログラミング関連スキル階層

文献[1]によるプログラミング関連スキル階層は，各スキルに対応する測定問題が使われた．そして試験結果をもとに回帰分析が行われ，図1のようなスキル間の相関を示した．なお図中の各スキルについては，次の小節で説明するが，例えば，コードを書くスキルは，Tracing2（繰返し構造があるプログラムのトレーススキル）と Explain（プログラムの内容を説明するスキル）と，正の相関が強いことを表している．

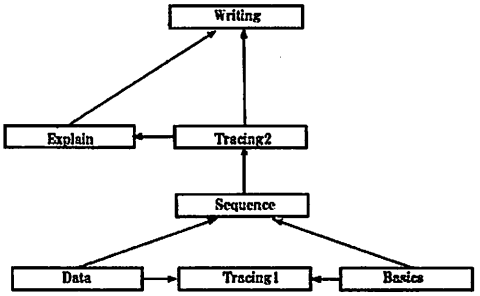

図 1 プログラミング関連スキル階層

Figure 1 Hierarchy of skill related to programming

2.2.2 文献[1]のスキル測定問題の概要

文献[1]で実施した Java によるスキル測定問題の内容について説明する．なおこの節の問題に関する説明，問題例は，全て文献[1]より引用した．

(1) **Basics(Question1-4)**

Java の構造を理解，一般的な Java 用語の定義とシンタックスエラーを発見する問題である．

(2) **Sequence(Question5-6)**

問題5は，命令行の一部が空欄となった空欄穴埋め問題である．そして，図2の問題6（Persons パズル）は，全てのコードを正しい順番に並べる問題である．問題6は，問題5の極端な場合である．

(3) **Tracing1 : non-iterative(Question7A-C)**

繰返し構造のないトレース問題である．例えば，初期化後の3つの代入文a+=2;b=4;c=b*aによって変更された3つの変数値を求める問題などである．

(4) **Tracing2 : iterative(Question7D-E)**

繰返し構造を含むトレース問題である．図3は，返り値の値を答える問題である．図4は，繰返し構造の中に"if"を含む場合で，返り値の値を答える問題である．

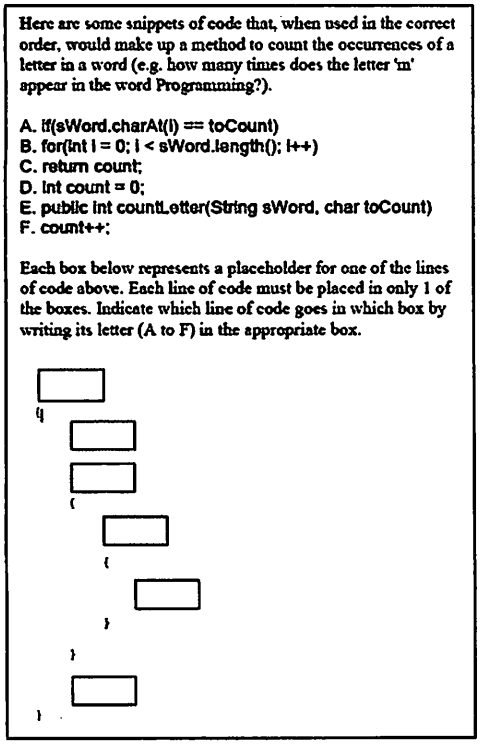

図 2　問題 6，Persons パズル [1]

Figure 2　Question6,the Parsons Puzzle [1]

```
public int q7D(int iLimit)
{
        int iIndex = 0;
        int iResult = 0;

        while (iIndex <= iLimit)
        {
                iResult += iIndex;
                iIndex++;
        }
        return iResult;
}
```

図 3　問題 7D，while ループ [1]

Figure 3　Question 7D,a while loop [1]

```
public int q7E(int[] aNumbers)
{
    int iResult = 0;

    for(int iIndex=0; iIndex<aNumbers.length; iIndex++)
    {
        if(aNumbers[iIndex] > iResult)
        {
            iResult =aNumbers[iIndex];
        }
    }
    return iResult;
}
```

図 4　問題 7E(i)，ループに"if"を含む [1]

Figure 4　Question 7E(i),a for loop containg an "if" [1]

(5)　**Exceptions(Question 7E ii ,12)**

例外に関する理解を問う問題である．例えば図 4 のコードで，q7E(null)と呼ばれた場合に起こることを答えるという問題である．

(6)　**Data(Question8)**

5 つのデータ型(ArrayList,Boolean,double,int,String)と「学生名」「結婚 or 未婚」などのデータ表現を結びつける問題，データ型を答える問題，変数宣言や型，scope の理解に答える問題である．しかも，コード中に小さなエラーがあって，コンパイルしても期待通りに動かない原因を答える問題も含んでいる．

(7)　**Writing(Question9,11)**

例えば，Java メソッドで，hip hop 名を生成して書くという問題である．学生は，指定されたタスクを 1 行のコードで書くことを求められた．

(8)　**Explain(Question10)**

コードが与えられ，それが何なのか平易な英文で説明する問題である．

```
public double method10A(double[] aNumbers)
{
    double num = 0;

   for(int iLoop = 0; iLoop < aNumbers.length; iLoop++)
   {
      num += aNumbers[iLoop];
   }
   return num;
}
```

図 5　問題 10A のコード，読解問題 [1]

Figure 5　the code for Question10A, a reading question. [1]

```
public void method10B(int iNum)
{
   for(int iX = 0; iX < iNum; iX++)
   {
      for(int iY = 0; iY < iNum; iY++)
      {
         System.out.print("*");
      }
      System.out.println();
   }
}
```

図 6　問題 10B のコード，読解問題 [1]

Figure 6　the code for Question10B, a reading question. [1]

(9)　**General(Question13)**

1 つのトピックスを選んで，4～5 行の明瞭な英文で説明を記述する．

3. プログラム修正・改造スキル

3.1 用語の定義

本論文では用語を以下の意味で用いる.

サンプルプログラム

プログラムの文法や命令の意味や書式を理解するために学習者に例題として示されるプログラムコードのことである. 例えば, 授業時に内容を説明する時などに使われる.

サンプルプログラムの類題

サンプルプログラムで使われた文法や命令の意味や書式と類似したプログラムの意味で使う. 機能の異なりについては, サンプルプログラムの一部を修正するだけで(サンプルプログラムのパターンマッチング程度で) 完成するような程度とする.

3.2 プログラム修正・改造スキル

プログラム修正・改造スキルを提案し, 用語の定義を説明する.

(1) プログラム修正・改造スキルの定義

本論文で取り上げる「プログラム修正・改造スキル」とは,「示されたサンプルプログラムを修正・改造して, サンプルプログラムとは異なる機能を有するプログラムを作成する能力のこと」と定義する.

文献[1]で Tracing を Tracing1 (繰返しのない構造) ,Tracing2 (繰返し構造) と分類しているように「プログラム修正・改造スキル」も複数レベル考える.「プログラム修正・改造スキル1」は, 繰返しがない構造のプログラムの修正・改造とし,「プログラム修正・改造スキル2」は, 繰返し構造があるプログラムの修正・改造と分ける.

(2) 本研究で対象外のプログラム修正・改造スキル

サンプルプログラムを修正したものであっても, 以下の場合は, サンプルプログラムの一部を修正するだけで完成する程度ではないので, 対象としない.

1) 例えば, 線形探索プログラムを2分探索に書き換えることで速度の改善を図るようなもの.
2) ソースコードの見た目の改善
3) コンピュータ資源の節約
4) 誤操作に対応した機能追加

3.3 プログラム修正・改造スキル測定問題

修正レベルについては, 例えば図 7, 図8のようにある程度の類推で解けるようなレベル, 図9のようにサンプルプログラム全体を理解しないと解けないレベルの複数レベルを用意する. これは, 被測定者の層によって, 多数が満点又はゼロ点となってしまうことで, 分析結果が得られない可能性を防止するためである.

(1) 「プログラム修正・改造スキル1」の問題例

サンプルプログラムの詳細が理解できなくとも, プログラム修正・改造点が類推できてしまうパターンである. 繰返し構造がない問題で, 特に Tracing1, Writing との相関に注目している.

```
Q1401. 設問の[改造点]となるように, 次のサンプルプログラムを修正してください.

[サンプルプログラム]入力された整数の最下位けたを除いた値を表示します.
[行番号]
1  import java.util.Scanner;
2  class Q1401{
3      public static void main(String[] args){
4          Scanner stdin = new Scanner(System.in);
5          System.out.print("整数入力:");
6          int a = stdin.nextInt();
7          int b = a / 10;
8          System.out.println("最下位けたを除いた値は" + b );
9      }
10 }

設問
[改造点]最下位桁を表示する.          [改造後の実行例]
                                    >java Q1401
                                    整数入力:234
                                    最下位桁:4
```

図 7 プログラム修正・改造問題

Figure 7 Program correction and remodeling problem

(2) 「プログラム修正・改造スキル2」の問題例

サンプルプログラムの詳細が理解できなくとも, プログラム修正・改造点が類推できてしまうパターンである. 繰返し構造がある問題で, 特に Tracing2, Writing との相関に注目している.

```
Q1402. 設問の[改造点]となるように, 次のサンプルプログラムを修正してください.

[サンプルプログラム]
1から10までの整数を加算して表示します.

[行番号]
1 class Q1402{
2   public static void main(String[] args){
3       int sum = 0;
4       for(int a = 1; a<=10; a++){
5           sum += a;
6       }
7       System.out.println("sum=" + sum);
8   }
9 }


設問
[改造点]1から10までの整数で奇数だけを加算する.

[改造後の実行例]
>java Q1402
sum=25
```

図 8 プログラム修正・改造問題

Figure 8 Program correction and remodeling problem

(1) 「プログラム修正・改造スキル2」の問題例

図9の例は, サンプルプログラムの詳細が十分に理解できないと, 修正が困難なパターンである. サンプルプログラムを理解するために必要になるスキル Tracing2 との強い正の関連を予想している.

01409. 設問の[改造点]となるように，次のサンプルプログラムを修正してください．

[サンプルプログラム]
"*"でピラミッド型の図形を表示します．

[行番号]
```
1  class 01409{
2     public static void main(String[] args){
3        int i, j;
4        int y = Integer.parseInt(args[0]);
5        for ( i = 1; i <= y; i++ )
6        {
7           for ( j = 1; j <= y - i; j++ )
8           {
9           System.out.print( " " );
10          }
11          for ( j = 1; j <= ( i * 2 - 1 ); j++ )
12          {
13          System.out.print( "*" );
14          }
15          System.out.println();
16       }
17    }
18 }
```

設問
[改造点]プログラムで出力される図形を 180 度回転して表示されるようにプログラムを修正してください．

[改造後の実行例]
```
>java 01409 4
*******
 *****
  ***
   *
```

図 9　プログラム修正・改造問題

Figure 9　Program correction and remodeling problem

4. 実験計画

4.1 方法

具体的には次のような実験を計画している．

(1) 概要

測定に使用する Java の問題セットは，原則として文献[1]に基づいて作問したが，プログラム修正・改造スキルは，提案スキルなので新たに作問した．試験問題セットは，問題の不備や難易度の調整などを数次の検討を行った．また測定は，実施時期を学習開始からそれほど差が出ないような日程を考慮して行うこととした．

(2) 測定対象

被測定者は，高等教育機関の大学，情報系専門学校での学習者で Java の基本知識を習得したレベルとする．なるべく多数で多様な教育機関のデータが集められることが望ましいことは云うまでもないが，個人で集めるのには，限界がある．そこで，Web を利用したデータ収集も試行することを考えている．この場合は，多様なレベルの学習者から多くのデータを収集できることが期待される反面，集まったデータの信頼性を高めることが必要になる．

(3) 測定問題の難易度

本研究の測定問題では，与えた各設問の成績に違いが無い場合は分析ができない可能性がある．例えば，学習者のプログラミング能力が非常に高い場合は全ての設問で満点になるかも知れないし，逆に能力が非常に低い場合はほとんどの設問で零点になるかも知れない．それを避けるためには，学習者に応じた適切なレベルの設問を用意する必要があるがこれは容易ではない．そこで，能力が異なると予想される様々な学習者のグループに対して，注目するスキルではスキルごとに難易度の異なる複数レベルの問題からなる同一の設問を使うことにする．

(4) 測定結果の分析

提案スキルと文献[1]のスキル間の相関分析を行う．分析は，教育機関や測定日の違いなどによる実験群ごとに行い一般的な平均点や分散とスキル間の相関を示す指標から分析を行う．また実験群ごとの測定値の傾向や相関などから，新たな提案ができるものと期待している．

4.2 分析方法

(1) 重回帰分析

プログラミング関連スキルの階層化は，スキル別に用意された設問ごとの成績を因子分析することで，スキル間の相関を明らかにする．特に「プログラム修正・改造スキル」と文献[1]で提案されている各スキルとの関係を分析する．

回帰分析は，原因となるデータ（説明変数）から結果(目的変数)を推定する回帰式を求めるものである．ある説明変数が，目的変数に有意な影響を与えているかどうかを知ることができる．重回帰分析は，説明変数が２つ以上ある回帰分析である．例えば，目的変数：Writing スキル，説明変数：Explain，Tracing2，プログラム修正・改造スキルとして分析を行う．

(2) 変数選択法

目的特性を説明するのに役立つと考えられる変数 x1,x2, … ,xp があって，p の数が 10 以上もあるときには，その全部を用いた重回帰式を求めるのではなく，その一部の変数だけを用いた重回帰式で十分間に合う場合が多い．すなわち，変数の数を少なくすると，重回帰が簡単になるだけでなく，その固有技術的な解釈がしやすくなるという利点がある．このとき，「p 個の変数のどのような部分集合として選ぶべきか」という変数選択の問題がある．変数選択法は，説明変数の選択方法に，目的変数に大きく影響する説明変数を選択して回帰式を求める．[2]

4.3 実験から予想するスキル階層（仮説）

定性的な分析から仮説を立てることを試みる．

(1) 定性的分析

ある一部の学習者を見ての定性的な分析である．

- プログラムを十分に細かく理解していなくても，類推で修正・改造してしまう学習者がいたので，Tracing2 と強い正の相関，Tracing1 と弱い正の相関があると予想する．
- 基本的な文法，命令書式が理解できないと，修正・改造できないと考えられるが，類推によってある程度対応できるので，Basics, Data と弱い正の相関があると予想する．

もし，プログラミングの学習中に一定間隔で連続して実験データを取れるような実験ができれば，次の予想についても明らかにしたい．

- プログラム修正・改造スキルを養成する演習を時間かけて行うとコードが書けるようになる学習者もいるので，Writing と弱い正の相関があると予想する．

(2) **仮説**

プログラム修正・改造演習は，文献[1]の Basics と Tracing1 の養成にも当たるが，特に Tracing2 の養成にあたるので，強い正の相関がある．また Writing とは弱い正の相関であると予想している．この仮説を図 1 の文献[1]のスキル階層に書き加えたものが，図 10 である．

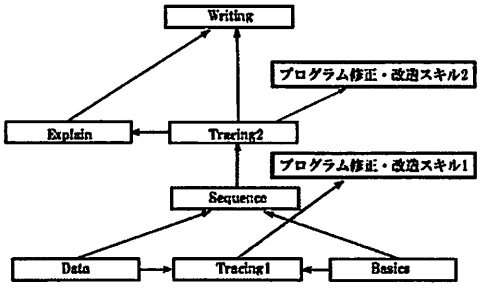

図 10　プログラミング関連スキル階層（仮説）
Figure 10　Hierarchy of skill related to programming (Hypothesis)

(3) **図 10 から予想される点**

図 10 から次のような現象が起こると予想されるので，検証できるような測定問題の作成とデータ収集に努めたい．

- プログラム修正・改造演習は，Tracing2 スキルの養成の一方法であると考えられる．よって，プログラム修正・改造演習だけの学習では，直接的な Writing スキルの養成にならない．

5. まとめ

本論文は，スキル間の相関を分析してスキル階層を示した文献[1]にもとづいて「プログラム修正・改造スキル」を提案した．そして，「プログラム修正・改造スキル」がプログラミング関連スキル階層のどの位置にあるかを得るための研究デザインを示した．実証実験計画では，プログラミング関連スキルの相関やスキル階層について研究を行う実験計画やデータ分析方法を説明した．最後に，実験結果を予想し仮説を立て，予想される点について述べた．

プログラム修正・改造スキルを含むプログラミング関連スキル階層の知見を得る研究によって，コードが書けない学習者に対して，個々の学習者がつまずいているスキル階層を知ることができるようになると考えている．このことから個々の学習者は，どのようなスキルを重点的に練習すればコードが書けるようになるかを知ることができるようになるので，コードが書けない学習者に対して，最適なプログラミング学習の方法を提供する知見が得られることを目指している．

今後の課題としては，各スキルを的確に測定できるような問題の確立と，多様な教育機関で学ぶ学習者の多くの測定データが得られるかどうかである．この点に関しては，個人で実施できる測定には制約や限界があるので，広く客観性のあるデータが収集できる仕組み作りも模索したい．また，プログラム関連スキルには，今回提案した以外にも多様なスキルが考えられる．例えば，デバッグ，プログラミングテスト，システムデザインなどである．今後これらのスキルについても研究の意義を考えながら，対象を広げていきたい．

参考文献

1) Mike, L. and Jacqueline, W. and Phil, R. and Raymond,L.: Relationships between reading, tracing and writing skills in introductory programming, ICER '08 Proceeding of the fourth international workshop on Computing education research, (2008).
2) 奥野忠一，久米均，芳賀敏郎，吉澤正: 多変量解析法，日科技連出版社(1983).