

# プログラミング素養診断テストの有用性の検証

北英彦\*1・寺久保丞\*2

Email: kita@elec.mie-u.ac.jp

\*1: 三重大学工学研究科電気電子工学専攻

\*2: 三重大学工学部電気電子工学科 (2016年3月卒業)

◎Key Words 素養診断, プログラミング, プログラミング能力

## 1. はじめに

現代社会においてプログラミングのスキルは理系だけでなく、文系の学生にも重要視されている。しかし、プログラミングは学習者の素養によって理解度に大きく差が出てしまい、適切な教育を行うことが難しい。

この問題に対して小林は、プログラミングで用いる概念を抽象化した問題によってプログラミングについて学ぶ前にプログラミングの素養を測る手法<sup>[1]</sup>を考案した。この手法を用いることでプログラミングの素養を評価し、プログラミング未経験者に適切な学習環境を提供することを目的としている。素養診断テストの検証実験を実施した結果、プログラミング関係の科目の評価とは相関がみられたが、検証実験の対象は本来の用途として想定しているものとは異なり全員プログラミング経験者であった。このため適切な結果が得られなかった可能性がある。

本研究では、プログラミングの未経験者を対象に検証実験を行い、素養診断テストの有用性を検証する。

## 2. 素養診断テストの内容

プログラミングの学習に必要な「決められたルールを正しく使って問題を解く能力」を評価するため、プログラミング学習者がつまづきやすく、かつプログラミングの基本的な概念である「代入とシーケンス実行」「分岐実行」「繰り返し実行」の問題を出題する。

問題で用いる式や文の表現は、プログラミング言語、数学、電気電子工学科の専門科目で使われているものではなく、上記のプログラミングの基本的な概念に対して、対象者にとっては初見と思われる、かつ、短時間で理解可能なものを新たに作成してそれらを用いた。

### 2.1 代入とシーケンス実行の問題

代入とシーケンス実行のテストでは、図1に示す箱に数値を格納する命令と、それらの命令を用いた箱同士の内容を入れ替える命令を用いて上から順に逐次処理を行う問題を出題する。プログラミングを学ぶ上で最初に理解すべき基本的な概念である変数および逐次実行を抽象化した問題である。

### 2.2 分岐・繰り返し実行の問題

分岐実行・繰り返し実行を理解する素養があるかどうかのテストとして、図2(a)に示す演算を図2(b)の中で分岐実行として示される順序に従って処理する文

を用いた問題を出題する。

プログラミングを学ぶ上で次に理解すべき基本的な概念が分岐と繰り返しである。

プログラムの実行の流れの基本的な制御構造は、逐次実行、分岐、繰り返しであり、これらの概念と学習するプログラミング言語におけるそれらの表現を理解し実際に使用することができれば基本的なプログラムは作成することができる。

本研究の素養診断テストは、これら三つの基本的な概念を学習者にとっては初見の表現で与えることで、プログラミング言語を学ぶことに対する素養の有無を測ろうとするものである。

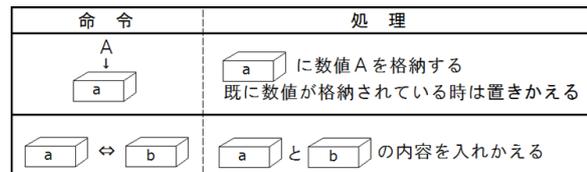


図1 代入とシーケンス実行の命令の定義

A ⊙ B : A, B の和を表す 例: 3 ⊙ 5 = 8, 7 ⊙ 5 ⊙ 10 = 22
A ⊕ B : A, B の差の絶対値を表す 例: 3 ⊕ 5 = 2, 7 ⊕ 5 ⊕ 10 = 8
A ○ B : A, B の差のうちの最大の値を表す 例: 3 ○ 5 = 5, 7 ○ 5 ○ 10 = 10

(a) 演算の定義

処理順序: 基本は左から順に計算
分岐実行 [   ] : 直前の結果が偶数のときは   の左を, 奇数のときは右を実行 例: 2 [ ⊙ 2   ⊕ 2 ] = 4 2 ○ 3 [ ⊙ 1   ⊕ 1 ] = 2
繰り返し実行 【   】 : 【   】内の   の左の演算を   の右で指定された回数繰り返す 例: 3 【 ⊙ 2   ⊕ 3 】 = 3 ⊙ 2 ⊙ 2 ⊙ 2 = 9

(b) 処理順序の命令の定義

図2 分岐実行・繰り返し実行の命令と処理順序の定義

### 3. 検証

#### 3.1 検証実験の対象

三重大学工学部電気電子工学科の科目であるプログラミング演習Ⅰ・Ⅱ 2015年度<sup>[2][3]</sup>の受講者であり、かつ、再履修者は含めずに今回の研究対象であるプログラミング未学習の学生76名を対象として検証実験を行った。

#### 3.2 素養診断テストの実施

対象の学生に対して本研究の素養診断テストを三重大学が使用しているe-ラーニングシステムのMoodleの小テスト機能を用いて実施した。出題した問題の種類と数は以下のようにした。

- 代入とシーケンス実行の問題 4題
- 分岐実行の問題 4題
- 繰り返し実行の問題 4題
- 分岐実行と繰り返し実行を組み合わせた問題 2題

以上の計14題を制限時間15分で実施した。また、1問1点の14点満点とした。問題の種類によって異なる重みをつけることも考えられるが、重みを付ける目安がないため、今回は全て同じ重みとした。

#### 3.3 検証の方針

本研究は、プログラミングの未経験者がプログラミングの科目を受講する前の時点で、彼らの中からその素養がない者を検出することを目的としている。プログラミングの素養のない学習者を事前に検出できれば、プログラミング演習のときに注意を払うことで適切な指導することが期待できる。

小林が考案した手法の有用性を検証するために素養診断テストの結果と以下の項目との相関を調べた。

- プログラミングの科目の評価
- プログラム作成の課題の評価
- 他の科目の評価（一般的な学習能力を測っているのではないことを確認するため）

### 4. 検証実験の結果

#### 4.1 プログラミングの科目の評価との相関

素養診断テストの結果と科目「プログラミング演習Ⅰ」の素点との相関を図3に示す。素点は、小テスト、プログラム作成の課題、定期テストなどの得点を積算したものである。相関係数0.32で、「やや相関がある」という結果が得られた。なお、相関の強さに関する表現は「図解でわかる統計解析」に基づいている<sup>[4]</sup>。相関はExcelを用いた簡単なツールを作成して求めた。

表1 相関係数の値と相関の強さ

相関係数の値	相関の強さ
0 ~ 0.2	ほとんど相関がない
0.2 ~ 0.4	やや相関がある
0.4 ~ 0.7	かなり相関がある
0.7 ~ 1.0	強い相関がある

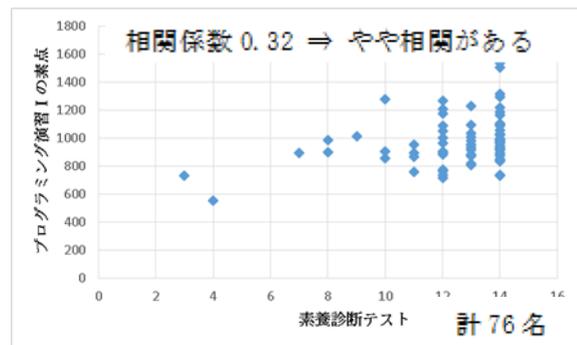


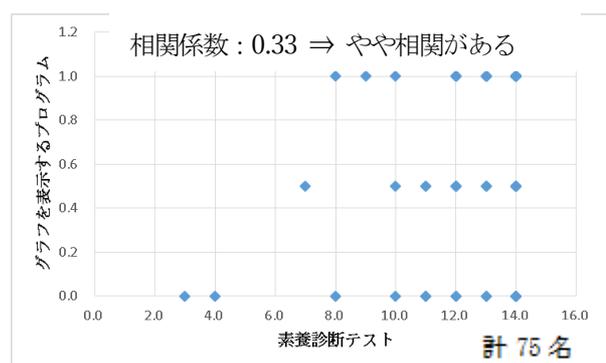
図3 科目「プログラミング演習Ⅰ」の評価との相関

#### 4.2 プログラム作成の課題の評価との相関

科目の評価には受講態度や努力点のようなプログラミングの能力とは直接関係ないものも含まれている。そこで、素養診断テストの結果と「プログラミング演習Ⅰ・Ⅱ」で行ったプログラム作成の演習課題の全54問中出来不出来の差が大きい代表的な4問との相関を調べた。その一例を図4に示す。検証の結果、4問中どの演習課題についても相関係数が0.4を上回ることはなく、「ほとんど相関がない」もしくは「やや相関がある」ということがわかった。なお、講師による採点だとプログラミングスタイルなどの不適切さで減点されてしまうため今回はプログラムの動作が正しいかのみを3段階で採点した。今回使用した採点基準を表2に示す。

表2 採点の基準

点数	プログラムの内容
0.0	全然できていない
0.5	半分はできている
1.0	ほぼできている



(a) グラフを表示するプログラム

図4 プログラミングの演習課題の相関

素養診断テストと演習課題の間にあまり相関がみられなかったのは、素養診断テストが簡単な問題で構成されているため図5に示すように大半の学習者が満点もしくは満点に近い得点を取っており、得点にバラつきがないためだと思われる。

そこで、素養診断テストはプログラミングの素養がない学習者を事前に選別することを目的としているので、素養診断テストの得点が低い学生のみを対象とし

て素養診断テストの結果とプログラミングの演習課題の評価との相関の有無を調べた。

素養診断テストの得点が11点以下の学生を対象にしたときの相関を図6に示す。全学生を対象にしたときと同様に「ほとんど相関がない」もしくは「やや相関がある」という結果になり、大きな変化はみられなかった。

次に、素養診断テストの得点が10点以下の学生を検証対象にしたときの相関を図7に示す。今回比較対象にした全てのプログラミング課題において「かなり相関がある」という結果になり、全学生を対象にしたとき及び素養診断テストの得点が11点以下の学生を対象にしたときよりも相関が強くなることがわかった。

素養診断テストの得点が低い学習者はプログラミングが得意でないことが分かる。ただし、素養診断の得点が比較的高い学習者であってもプログラムの作成ができていないものもいるので、素養診断テストでプログラミング能力の素養を判別できているわけではない。

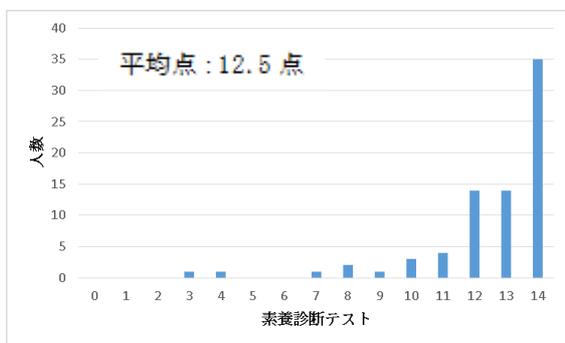
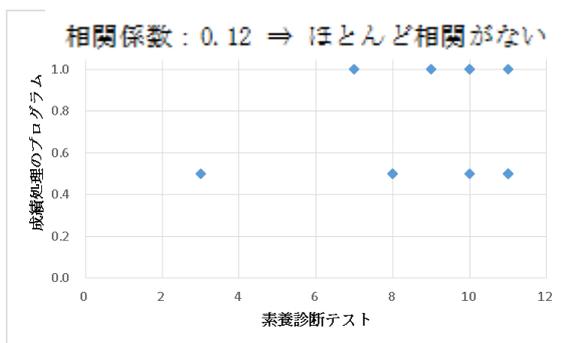
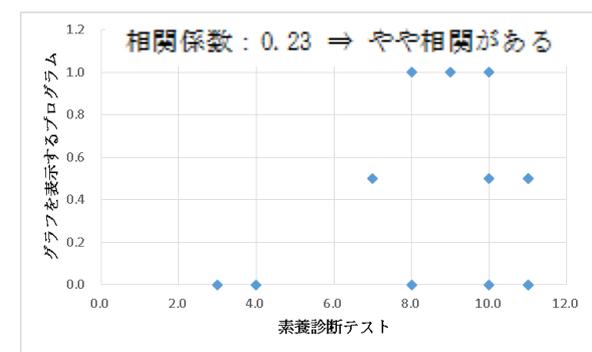


図5 素養診断テストの得点の分布

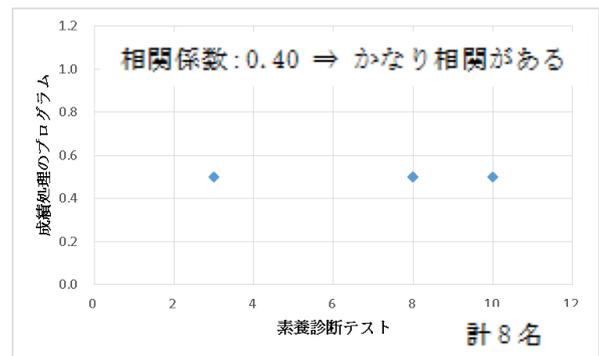


(a) 成績処理のプログラム

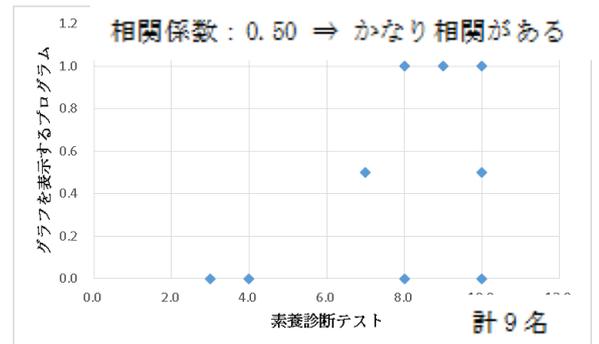


(b) グラフ表示のプログラム

図6 素養診断テストの得点が11点以下の学生



(a) 成績処理のプログラム



(b) グラフ表示のプログラム

図7 素養診断テストの得点が10点以下の学生

表3 素養診断テストと演習課題の相関

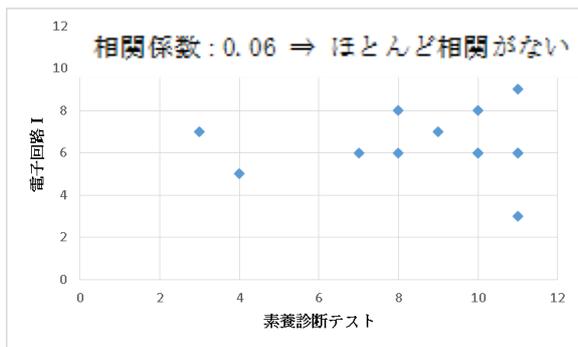
対象の学生	プログラミング課題	
	成績処理	グラフ表示
全員	0.15 ほぼ相関なし	0.33 やや相関あり
素養診断テスト11点以下の学生	0.12 ほぼ相関なし	0.23 やや相関あり
素養診断テスト10点以下の学生	0.40 かなりの相関	0.50 かなりの相関

### 4.3 他の科目の評価との相関

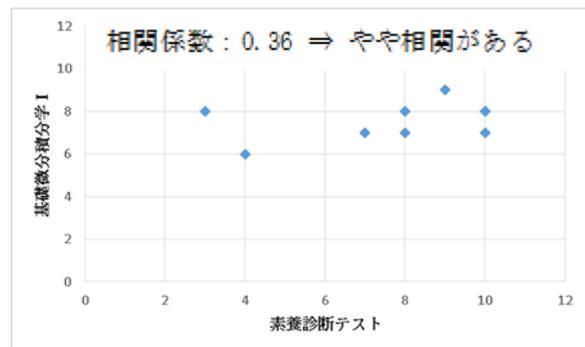
素養診断テストは簡単な知能テストのような問題なので素養診断テストの得点が低い学習者はそもそも学習能力が低い可能性がある。そこで、素養診断テストの得点が11点以下の学生を対象にして素養診断テストの結果と電気電子工学科で開講している数学や電気回路、および、教養教育の未習外国語など学習者が履修済みの代表的な9科目の評価を比較して相関の有無を調べた。

その結果を図8に示す。必ずしも一般の学習能力とは相関があるわけではなく、「ほとんど相関がない」ものから「強い相関がある」ものまで幅広い関係がみられた。

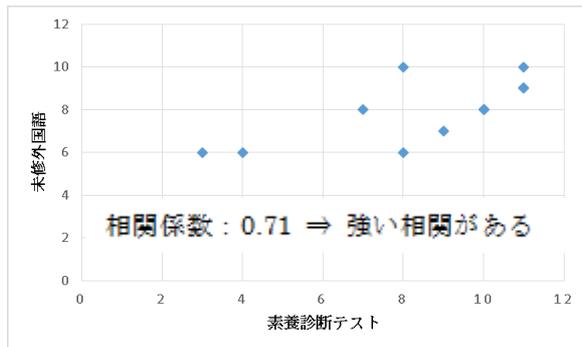
「未修外国語」の評価との相関係数は0.71となり唯一「強い相関」が得られた。この理由を考察すると、裏付けとなる調査は行っていないが、プログラミングも言語の一種であり、どちらも同じ未学習の言語という点では類似しているためだと思われる。



(a) 「電子回路 I」の評価との相関

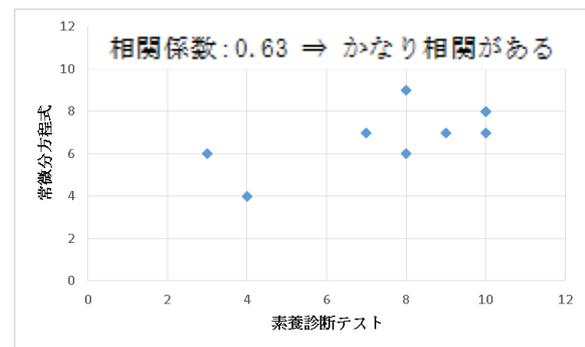


(a) 「基礎微分積分学 I」の評価との相関



(b) 「未修外国語」の評価との相関

図 8 素養診断テストの得点が 11 点以下の学生の点数と「他の講義の成績」の相関



(b) 「常微分方程式」の評価との相関

図 9 素養診断テストの得点が 10 点以下の学生の点数と「他の科目の成績」の相関

図 9 にその一部を示すように素養診断テストの得点が 10 点以下の学生も対象にして各科目の評価との相関の有無を調べた。その結果、「強い相関がある」ものはみられなかったが、同様に幅広い関係がみられたので、こちらも必ずしも一般の学習能力とは相関があるわけではないことがわかった。

## 5. 結論

プログラミング未学習者を対象に検証実験を行い素養診断テストの有用性を検証した結果、大多数が素養テストで高得点を得ているため、相関があまりみられなかったが、素養診断テストの得点が低い学習者のみを対象にして検証実験を実施すると相関がみられたのでプログラミングの素養のない一部の学習者の選別に成功していることがわかった。

しかし、素養診断テストの得点が高いにもかかわらず、プログラムを上手く作成できていないものも存在しているため、プログラミングの素養を評価する手法としては不十分である。

## 6. プログラミング演習での利用の仕方

プログラミング演習の科目での素養診断テストの結果の利用の仕方のひとつは能力別クラス編成である。ただし、素養診断テストは簡単な問題であるため得点の低い学習者は少ないためひとつのクラスを設けるほどではない、また、素養診断テストの信頼性が十分でないこともありプログラミングの学習の前に苦手だと学習者をラベル付けするのも問題がある。他には、演習中の講師や TA の机間巡回のとき

に指導に行きやすくするために教室内の座席を配慮することができる。また、我々が開発および運用をしているプログラミング演習システム PROPEL<sup>[5][6][7]</sup>の講師用画面において素養診断テストの得点が低い学習者には印をつけておくことを考えている。

## 7. まとめ

得点が低い学習者を対象にするとプログラムを作成する能力との間に相関がみられたので、本研究の素養診断テストは明らかにプログラミングが苦手な学習者の選別は可能なのでこの点では有用性がある。

## 参考文献

- [1] 小林史生, 北英彦: 学生のプログラミングの素養を調査する手法, コンピュータ利用教育協議会, CIEC 春季研究会 2015 (2015)
- [2] 三重大学ウェブシラバス (2015 年度) プログラミング演習 I, <http://syllabus.mie-u.ac.jp/?action=display&id=19567>
- [3] 三重大学ウェブシラバス (2015 年度) プログラミング演習 II, <http://syllabus.mie-u.ac.jp/?action=display&id=19574>
- [4] 前野昌弘, 三國彰: 図解でわかる 統計解析, 日本実業出版社 (2000)
- [5] 戸上稔磨, 北英彦: プログラミング演習のためのプログラムテスト支援機能, 2016PC カンファレンス (発表予定) (2016)
- [6] 上村拓磨, 北英彦: プログラミングスタイル習得のための自己学習環境, 2016PC カンファレンス (発表予定) (2016)
- [7] 四方雅晴, 北英彦: プログラミング演習システムにおける音と色を用いたユーザインタフェースの改善, 2016PC カンファレンス (発表予定) (2016)