学習者を多面的に評価する
プログラミング初級者向け教育支援ツール

橋浦研究室　　1125319　高橋　匡平　　1125361　濵野　智史

１．はじめに

ソフトウェアはプロジェクト形式で開発される.プロジェクトでは多人数が作業を分担して行うため,他人が自分のソースコードを利用したり,自分が他人のソースコードを利用することは避けられない.その一方，ソフトウェアエンジニアを養成する過程では,求められた機能を満たしているかどうか（機能性）が重視され，多人数が参加するプロジェクトでどのようなコードを書いたら良いのかを学習する機会は限られている.

２．ソフトウェアの品質

ソフトウェアの品質には様々な特性が含まれており，それらを評価する国際規格であるSQuaRE System and software quality models[1]では，品質特徴を８つの特性に分類している.このうち,他人数が参加するプロジェクトにおいて，特に留意しなければならない特性として，本研究では保守性に着目する． 保守性が高いソフトウェアは,他人がソースコードを読んだ時に誤解や修正ミスを生みづらく,バグが入り込む可能性を低くすることに繋がるためである.この保守性にはさらに５つの副特性が定義されており,前述の留意点に照らして修正性に着目することとした．修正性の定義は以下に示す通りである.

- 4.2.7.4　修正性（modifiability）
  - 欠陥の取込みも既存の製品品質の低下もなく，有効的に，かつ，効率的に製品又はシステムを修正することができる度合い

３．研究目的

ソフトウェアを修正する際には元となるソースコードを読み込むことが必要となる.そのような時に読みにくいソースコードが１つでも混入していると，修正作業は困難になり，修正性は低下する.このようなソースコードの読みやすさは一般に可読性と呼ばれている.

可読性が高いプログラムが書けるようになるためには,専門的なトレーニングを受けることが望ましいが,前述のとおりこのような学習の機会は限られている.

このような状況に対し,本研究では他人が読むことを前提とした可読性が高いソースコードを書けるようにするための教育支援ツールの開発を行う．さらに，学習者の試験の解答結果を構文解析によって解析し,独自に決定したスタイルに基づいて採点を行う.加えて，その採点方法が妥当であるかについても確認する.

３．ソースコードのスタイル規約

本研究では可読性を構成する要素のうちインデントのスタイルに着目した.インデントのスタイルとは､プログラム内部の構造を明らかにするために,字下げや空白の位置をルール化したものである.これには様々な方法がこれまでに考案されているが,代表的なものとしてはK&Rスタイル[2]やBSDスタイル[3]がある.

本研究で採用したスタイルは文献[4][5]を参考として決定した．本研究で用いる，スタイルの例と適用した規約を以下に示す.

```
 6  int main(void)
 7  {
 8        int i;
 9        int test[NUM];
10        printf("テストの点数を入力してください。¥n");
11
12        for(i = 0; i < NUM; i++)
13        {
14              scanf("%d", &test[i]);
15        }
16
17        printf("最高得点は%d点です。¥n", max(test));
18
19        return 0;
20  }
```

図１　スタイルを適用したソースコード

〈スタイル規約〉（一部抜粋）

① 6行目：括弧を用いる際は，左括弧の後と右括弧の前に空白を入れない
   - 例：`main( void )`としない

② 7～20行目，13～15行目：中括弧は，関数定義や制御文など全て改行し,前の行と揃えたカラムに置く

③ 12 行目：演算子の前後には空白を入れる．
- 例：`i=0; i<NUM;` としない

④ 14，17 行目：同一行内で「,」の後にコードを書く場合，空白を 1 つ開けて書く
- 例：`int i, max;` としない

４．ツールの実装

本ツールの概要は，ウェブブラウザ上で学習者にプログラミングの穴埋め問題を出題し，回答させるものである．

ツールはプログラムの正誤の判定を行うと共にスタイル規約と整合しているかどうかの判定を行う．整合している場合には，正誤の判断のポイントに加えて，スタイル規約ごとに定められた追加点を与える．よって，学習者が高得点を取るには，問題に正解すると同時に，スタイル規約に準拠する必要があり，プログラミングの問題を解きながらスタイル規約について知識を深めることができる．

ツールは JSP を用いたウェブアプリケーションとして実装を行った．

５．実験と考察

ツールの有効性を確認するために、被験者にツールを用いてもらうことを２回繰り返し，点数を比較することで行った．実験の概要を以下に，結果を表１に示す．

- 被験者： 本学科橋浦研究室の 3 年生 8 名
- 問題数：17 問（Web サイト[6]の問題を元に作成）
- 追加点の採点方法：追加点は空白の入れ方に関しては 1 つ 1 点，インデントに関しては統一されていた場合 5 点，波括弧の位置は同じ行に書かれていて統一されていれば 3 点

表１より，８人中 5 人の被験者については追加点の向上が確認できた。効果が見られなかった被験者についてさらに詳細に解答内容を確認したところ，問題ごとにスタイルを変えたり，前回の結果から修正すべき内容を変えずに解答しているケースが見られた．

７．結論と今後の課題

本研究ではソースコードの可読性に注目し，そのスタイルを追加点として評価するシステムを構築し，評価実験により一定の有効性があることを確認した．

今後は学習者の知識量を測り，学習者ごとに問題の出題方法を変える「学習者モデル」を取り入れることで，解答者ごとに問題点の特定が可能になるため，問題に応じたメトリクスを設定することが出来ると考えられる．

参考文献

[1] ISO, ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models," http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35733 (2016/01/18 閲覧)

[2] B. W. Kernighan, D. M. Ritchie,"プログラミング言語 C," 共立出版，1981.

[3] The FreeBSD foundation,"Kernel source file style guide," FreeBSD Kernel Developer's Manual, http://www.freebsd.org/cgi/man.cgi?query=style&sektion=9 (2016/01/18 閲覧)

[4] 黒田康太，"ヒューリスティック C プログラミング-ビギナーズコース-," 東京電機大学出版局，1990.

[5] 独立行政法人情報処理推進機構，"【改訂版】組込みソフトウェア開発向けコーディング作法ガイド [C 言語版] Ver.2.0 (ESCR Ver.2.0)," オーム社，2014.

[6] ともじ,"初心者のためのポイント学習 C 言語," http://www9.plala.or.jp/sgwr-t/ (2016/01/18 閲覧)

表１ 実験結果

	1回目		2回目		変化	
被験者	正誤点	追加点	正誤点	追加点	正誤点	追加点
A	40/100	62/72	20/100	103/120	−20	+41
B	0/100	45/76	0/100	65/82	0	+20
C	0/100	51/65	0/100	39/75	0	−12
D	20/100	61/81	40/100	57/91	+20	−4
E	20/100	71/87	40/100	72/82	+20	+1
F	0/100	30/59	20/100	33/61	+20	+3
G	20/100	50/74	0/100	46/111	−20	−4
H	20/100	64/85	20/100	93/101	0	+29