

平成 30 年 6 月 12 日現在

機関番号：32643

研究種目：基盤研究(C) (一般)

研究期間：2015～2017

課題番号：15K01035

研究課題名(和文)Javaプログラミングスタイル学習のためのEclipseプラグインツール

研究課題名(英文)Eclipse Plugin Tool for Learning Programming Style of Java

研究代表者

荒井 正之 (ARAI, Masayuki)

帝京大学・理工学部・教授

研究者番号：70212602

交付決定額(研究期間全体)：(直接経費) 3,000,000円

研究成果の概要(和文)：本研究の目的は、Java言語のプログラミング初学者のために、プログラミングスタイルが自学自習可能なツールを開発することである。学習者は、このツールを用いることにより、(i)適切な字下げ、(ii)ソースコードに対する適切なコメントの挿入位置、(iii)Javaの命名規則に従った適切なクラス名、メソッド名、変数名、(iv)変数名に英語を用いて命名することの重要性、(v)適切な変数のスコープなどを学ぶことができる。本ツールは、多くのプログラマーが開発環境として利用している、Eclipseのプラグインとして実装した。ツールを授業に用いて評価した結果、有効性が認められた。

研究成果の概要(英文)：We developed an Eclipse plugin tool for learning the programming style of Java. The tool has the following functions: (1) recommends using CamelCase and English words for the names of classes, methods, and variables; (2) points out incorrect scope levels of variables and inappropriate lengths of variable names; (3) recommends writing comments in source programs; (4) shows sample source programs in the Java programming style. The evaluation results show the tool is efficient for learning the programming style of Java.

研究分野：情報科学分野の教育

キーワード：プログラミング プログラミングスタイル Java Eclipseプラグイン 学習支援

1. 研究開始当初の背景

プログラミングスタイルは、ソフトウェアの開発を効率的に行うために非常に重要である。プログラミングスタイルに従って、ソースプログラムを書くことにより、コードが読みやすくなるとともに、バグが減ると考えられる。プログラミング初学者は、一般にプログラミングスタイルに対する意識が低い。また、プログラミングスタイルが重要にも関わらず、現在のプログラミング授業では、プログラミングスタイルに関する教育がほとんど行われていない。そのため、プログラミングスタイルを身につけられない学生が多い。

プログラミングスタイルの学習支援に関する研究として、構文構造における不正なプログラミングスタイルを自動的に検出するシステム、プログラムにおける落とし穴を自動的に検出するシステムなどがある。これらは学習者が作成したソースプログラムに対して、予め指定した不正なプログラミングスタイルを検出することにより、プログラミング初学者でも不正なスタイルがチェックできる。また、これらは構文構造など部分的なスタイルに対応するものであり、プログラム全体のスタイルについての支援を行うことができない。一方、プログラム全体のスタイルを対象としたものとして、プログラム内のアクティブ変数を表示して、モジュール分割することで支援するシステムがある。Pascal 言語で書かれたプログラムの全体設計から、コメント、変数名の命名等に関する支援を行うことができる。

本研究で提案するツールは、Java 言語で書かれたプログラム全体に対して、インデント(字下げ)、コメントなどの支援を行うとともに、Java の命名規則に従ったクラス名、メソッド名、変数名の支援、またそれらの名前を英語でつけるように支援するものである。インデントやコメントに対する支援機能を持つシステムは多いが、命名規則や英語の命名に対する支援機能を持つシステムは新しい。

2. 研究の目的

本研究では、前述の問題点を解決するために、Java 言語のプログラミングスタイルを身につけさせるための支援ツールの開発を目的とする。

3. 研究の方法

図 1 に示すプログラムを提案ツールに入力した場合を例に、ツールの概要について述べる。図 1 は、次の点で Java のプログラミングスタイルに従っていないプログラムである。

- ・ 5 行目のクラス名 drawSomeMaru の頭文字が大文字になっていない。
- ・ 7~11 行目は、paint メソッドの中であ

るにも関わらず、インデントが 1 段深くなっていない。

- ・ 9~10 行目は、for ループ内にも関わらず、インデントがさらにもう 1 段深くなっていない。
- ・ 11 行目にプレイス({})が 2 つ書かれている。
- ・ コメントが 1 つも書かれていない。

図 1 に示す Java 言語のソースプログラムを、ツールに入力したときの実行結果を図 2 に示す。

画面左側には、Java 言語のプログラミングスタイルに従って、整形されたソースプログラムと、プログラミングスタイルに従っていないクラス名、メソッド名、変数名に対する警告等を表示する。また、画面左右には、「/* コココメ!! */」を挿入したソースプログラムを表示する。なお、「/* コココメ!! */」とは、「ここにコメントを入れなさい!」という意味で使用している。

ツールのユーザは、左側の手本を見ながら、画面右側のソースプログラムの訂正、画面右側のソースコードにコメントを書き入れること等により、Java 言語のプログラミングスタイルを学ぶことができる。なお、左側のソースプログラムは、コピーやペースト等の編集ができないようになっている。

```

1: import java.awt.*;
2: import java.applet.*;
3:
4:
5: public class drawSomeMaru extends Applet {
6:     public void paint(Graphics g){
7:         int size=10;
8:         for (int i = 0; i < 5; i++){
9:             g.drawOval( 10, 10, size, size);
10:            size = size + 10;
11:        }}
12: }

```

図 1 Java 言語のソースプログラムの例

4. 研究成果

(1) ソースプログラムから情報抽出

次章以降で述べる機能を実現するために必要な情報を抽出する。ツールに入力されたソースプログラムに対して、字句解析、構文解析を行った後、次の情報を抽出する。

- ・ プログラムの中で使われているクラス、メソッド、変数名の前と宣言されている行
- ・ 繰り返し処理、条件分岐処理等の開始行
- ・ インデントがずれている行
- ・ 1 行にプレイス({})が複数ある行 など

(2) キャメル方式推奨機能と実現方法

キャメル方式とは、読みやすいソースプログラムを書くために推奨されている Java の命名規約の 1 つである。クラス名をつける場合は、例えば ClassName のように「複合語

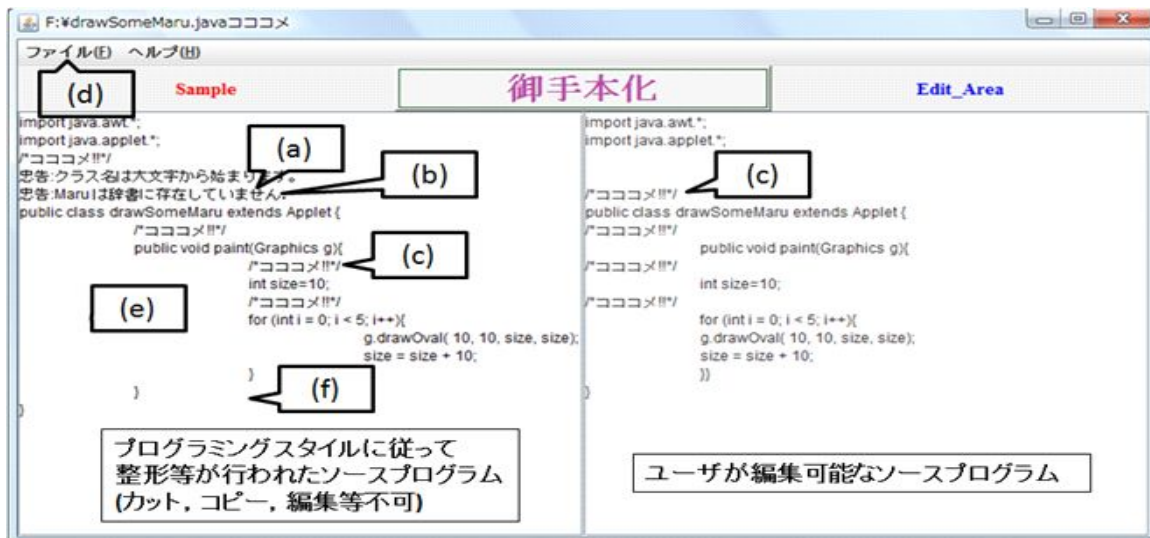


図2 図1のプログラムの実行結果

の場合は単語毎に先頭の文字を大文字にする」、「クラス名の1文字目は大文字にする」などが推奨されている。

ソースプログラムから抽出したクラス名に対して、1文字目を調べ、小文字の場合は図2(a)に示すように、該当箇所の前の行に「クラス名は大文字ではじめましょう」と警告する。

次にクラス名の先頭から文字を順に探索し、大文字の前でクラス名を分割する。例えば ClassName という複合語の場合、この処理により Class と Name という2つの単語に分割する。続いて、それらの単語が英単語辞書および英単語略語辞書にあるかを調べる。単語が辞書に存在しない場合、キャメル方式が守られていない可能性があるため、該当箇所の前の行に「英語辞書に存在しません」と警告する。例えば、クラス名が Classname などの場合、この警告が表示される。なお、英単語辞書および英単語略語辞書の検索方法については、次章で説明する。

一方、メソッド名、変数名に対する Java の命名規則は、例えば enlargeButton ように、「単語のつなぎ目の頭文字は大文字にする」はクラス名の規則と同じであるが、「1文字目は小文字にする」が異なる点である。これらの命名規則に従っていない場合や、英単語辞書と英単語略語辞書に合致する単語がない場合は、該当箇所の前の行に、「変数名は小文字ではじめましょう」、「英語辞書に存在しません」などの警告を表示する。

(3) 英単語推奨機能と実現方法

企業等では、プログラムを外国人と共同開発する機会が増えている。そのため、変数名等は外国人にも理解できるように英語にする習慣をつけることが望ましい。本章で述べる英単語推奨機能は、クラス名、メソッド名、変数名に対して、英語を使うことを推奨するための機能である。英単語が使用されていない場合は、図2(b)に示すように、該当箇所の前の行に「英語辞書に存在しません」と警告

する。例えば、図1のプログラムでは、クラス名が drawSomeMaru であるので、draw, some, maru の3つの単語が英単語辞書および英単語略語辞書にあるかを調べ、このうち maru が辞書にないため警告している。

英単語辞書および英単語略語辞書の検索には、Representational State Transfer (REST) [7]という手法を用いる。RESTは、XMLやhttpを使った簡易なwebベースのシステムで、URLを使用して様々な機能を利用することができる。イースト辞書 Web サービス[8]などのサイトで、英単語辞書検索のRESTが利用できる。英単語辞書検索用のRESTサービスでは、URLの中に検索したい単語を挿入することで、その単語の検索結果のXMLデータが返ってくる仕組みである。

図3に camel という単語を検索するときのURLの例、そして図4に返ってきたXMLデータの例を示す。TotalHitCount 1 /TotalHitCount の行が辞書にヒットした数を示している。図3では、camel という単語が辞書に1件登録されていることを表している。TotalHitCount が0の場合、図2(b)に示すように「単語が辞書に存在しません」と警告を行う。

同じ手法で英単語略語辞書のREST方式のサービスを利用することにより、英語の略語にも対応している。

<http://btonic.est.co.jp/NetDic/NetDicV09.asmx/SearchDicItemLite?Dic=EJdict&Word=camel&Scope=EXACT&Match=STARTWITH&Merge=AND&Prof=XHTML&PageSize=20&PageIndex=0>

図3 camel という単語を探索する URL

```

<?xml version="1.0" encoding="utf-8" ?>
- <SearchDicItemResult xmlns:xsi="http://www.w3.org/2001/
  <ErrorMessage />
  <TotalHitCount>1</TotalHitCount>
  <ItemCount>1</ItemCount>
- <TitleList>
  - <DicItemTitle>
    <ItemID>005804</ItemID>
    <LocID />
  - <Title>
    <span class="NetDicTitle" xmlns="">camel</span>

```

図 4 camel の検索結果

(4) その他の機能

・コメント付加推奨機能 3章で述べたように、ソースプログラムから、クラス名、メソッド名、変数名、繰り返し処理、条件分岐処理等の該当行を抽出する。該当箇所に対して、図 2(c)に示すように、画面左右のソースプログラムに「/*ココメ!!*/」を挿入する。

クラスの機能や目的、メソッドの振る舞い、変数の役割、繰り返し処理、条件分岐処理については、それらの処理の目的等のコメントを書く習慣をつけさせるための機能である。

・保存制限機能 図 2(d)に示す「ファイル」の中の「保存」が選択されると、ユーザが編集した画面右側のソースプログラムの「/*ココメ!!*/」の行が編集されたかチェックを行い、未編集の場合は保存することができない。半強制的にコメントを書かせるための機能である。

・ソースコード整形機能 前述のように、ソースプログラムから、インデントがずれている行、1行にブレイス({})が複数ある行を抽出する。該当箇所に対して、ソースプログラムの整形を行う。

インデントがずれている場合は、図 2(e)に示すように適切なインデントを行う。1行にブレイス({})が複数ある場合は、図 2(f)に示すように、ブレイスが1行に1つになるようにするとともに、適切なインデントを行う。読みやすいソースプログラムを書く習慣をつけさせるための機能である。

(5) ツールの評価

本ツールを使用後に本学の学部生 7 人に対してアンケートを行った。アンケートの内容は次のとおりである。

質問 1 このツールを使用して下記項目が理解できたか。

- ・適切なコメント位置
- ・キャメルスタイル
- ・適切なインデント
- ・変数名の重要性
- ・適切な変数名の長さ

質問 2 このツールを使って Java のプログラミングスタイルを学ぶことができるか。

質問 1 と 2 に対する集計結果を表 1 に示す。質問 1 の「コメントの位置」については全員が「ほぼ理解できた」以上であった。他の項目についてもほとんどの学生がほぼ理解で

きた以上と答えた。質問 2 についても 7 人中 5 人の学生が理解できたと回答した。上記の結果から、本ツールの有効性を確認することができた。

表 1 アンケートの集計結果

		理解できた	ほぼ理解できた	どちらとも言いえない	あまり理解できない	理解できない
Q1	コメントの位置	2	5	0	0	0
	キャメルスタイル	1	4	2	0	0
	インデントの位置	2	3	1	1	0
	意味のある変数名	2	3	2	0	0
	適切な変数名の長さ	0	5	1	1	0
Q2		2	3	2	0	0

5. 主な発表論文等

〔雑誌論文〕(計 1 件)

Masayuki Arai: Learning Support Systems for Novice Programmers, 117th International Conference on Education and E-Learning, pp.1-6 (2017.1) (査読有)
http://www.worldresearchlibrary.org/up_proc/pdf/653-14876710771-6.pdf

〔その他〕

ホームページ等

<http://www.ics.teikyo-u.ac.jp/lab/Arai/index.php?AraiLabHomePage>

6. 研究組織

(1) 研究代表者

荒井 正之 (ARAI, Masayuki)
 帝京大学・理工学部・教授
 研究者番号: 70212602