2J-05

初学者を対象としたプログラム中の識別子の適切性評価

川端 盛志　　玉田 春昭
京都産業大学

1 はじめに

プログラミングにするにあたり、識別子の命名は重要である。命名が適切にされていれば識別子からおおよその処理の予測が出来る。しかし、初学者には識別子の重要性が浸透しておらず、適切な名前を付けない場合が多い。その理由として、確固とした識別子の評価の基準が定まっていないためと考えられる。一方、経験者は、今までの経験から自分なりの評価基準が構築されており、名前の適切性が判断できる。

本稿では、初学者が自らつけた識別子の適切性の評価手法を提案する。提案する手法により、たとえ識別子の評価基準が定まっていない場合でも良し悪しを判断できるようになるため、適切な名前を付けることが期待できる。

2 提案手法

まず、既存のプログラミング演習科目で提出されたプログラムを対象に、初学者はどのような名前をつけるのかを傾向を確認する。そこから識別子を元にしたメトリクスを用いて初心者が名付けた名前の良し悪しを判定することとする。

2.1 提案メトリクス

次の２つのメトリクスでプログラムの識別子の良し悪しを判定する。

(1) 識別子につけられた単語が意味を持つか

(2) 解答例のプログラムでつけられている識別子にどの程度似ているか

それぞれ、メトリクスを定義するため、対象プログラム p から抽出した識別子の集合を $N(p) = \{n_1, n_2, ..., n_n\}$ とする。n_i が複数の単語から成り立っていた場合、それぞれの単語に分け、最終的に得られた単語集合を $N_w(p) = \{w_1, w_2, ..., w_m\}$ とする。一方、単語 w が辞書で見つかれば 1 を、見つからなければ 0 を返す関数 $d(w)$ を定義する。ここで、有意率 M を $M = \frac{\sum_{i=1}^{m} d(w_i)}{|N_w(p)|}$ と定義する。

(2) では、解答例のプログラム p_e と比較対象のプログラム p それぞれで定義されている識別子を比較する。それぞれから単語集合 $N_w(p_e) = \{w_i^e\}(1 \le i \le m_e), N_w(p) = \{w_j\}(1 \le j \le m)$ を抽出する。このとき、w を p から取り出したと比較し、シソーラス距離（概念距離）を求める。一番高い値を w のスコアとし、$d_i = sim(w_i, N_w(p_e))(1 \le i \le m)$ で求めるとする。このとき、p の名前の適正度 S を $S = \frac{\sum_{j=1}^{m} d_j}{|N_w(p)|}$ で求める。

ただし、適正度を測定するとき、本質的な名前にのみ着目したい場合も考えられる。そのため、適正度 S を拡張し、各単語のスコアでフィルタリングを行うことを考える。フィルタリングの閾値 t は与えられるものとしたとき、$N_d(p) = \{(w_1, d_1), (w_2, d_2), ..., (w_m, d_m)\}$ を得る。そして、この中から、$d_i \ge t$ となる要素のみを取り出す（$f(N_d(p)) = \{(w_i, d_i) \in N_d(p) | d_i \ge t\}(1 \le i \le m)$）。このとき、$f(N_d(p))$ から単語 w_i のみを取り出して $N_f(p)$ とし、距離 d_i のみを取り出して $D_f(p) = \{d_1, d_2, ..., d_k\}$ とする。そして、フィルタリングされた適正度 S' は $S' = \frac{\sum_{i=1}^{k} D_f(p)}{|N_f(p)|}$ とする。

2.2 識別子の良し悪しの判定

第 2.1 節で定義した２つのメトリクスを元に、初心者の書いたプログラムの良し悪しを判定する。２つのメトリクスはそれぞれプログラム単位で算出される。それぞれのメトリクス値が閾値よりも大きければ、良いプログラムであると判定する。

3 ケーススタディ

第 2.1 節で定義したメトリクスを実際のプログラムを対象に算出し、傾向を確認する。

3.1 対象データ

本ケーススタディの対象プログラムは、本学の 2 年生を対象にした Java プログラミング演習の課題である。本稿で対象にした 2013 年度の受講者数は 130 名であった。この演習では、Java の文法からオブジェクト指向について学習し、22 の課題が課せられる。このうち、提出率が 5 割以上の課題 10 個を対象に、1,036 個のプログラム（80 名の受講生）から識別子の抽出を行い、メトリクスを算出した。

Suitability Evaluation of Identifiers in Java Programs for Novice programmers
Seiji Kawabata　Haruaki Tamada
Kyoto Sangyo University

提案手法を実施するには辞書の設定が不可欠である。本稿では、WordNet を用いた [1]。

3.2 分析

3.2.1 分析データの概要

図 1 に課題ごとの平均ステップ数と提出数をグラフで示す。横軸が課題を表し、第 1 縦軸の棒グラフが平均ステップ数を表し、第 2 縦軸の折れ線グラフが提出数を表している。図からわかるように、課題 10 を含め全体的に非常に小規模なプログラムである。

3.2.2 課題ごとの有意率

図 2 に各課題で提出されたプログラムの有意率を箱ひげ図で表したグラフを示す。横軸が課題番号を、縦軸が各課題における提出されたプログラムの有意率を表している。課題が進むにつれ有意率の中央値は向上していくものの、課題 8、9、10 以外は、有意率の中央値が 4 割を下回り、全体的に有意率が低いことが分かる。

3.2.3 提案メトリクス (有意率，適正度) の分布

図 3 に提案メトリクスの分布グラフを示す。横軸は課題 10 の各プログラム 80 個、縦軸はメトリクスの値である。なお、有意率でソートしている。識別子を調査したところ、解答例の識別子は、識別子を見るだけで、動作を予測できるものの、課題では識別子から動作や内容の予測が難しい名前であった。

次いで、図 3 に示される有意率が一番高いプログラムと、一番低いプログラムを調査した。有意率が最高のプログラムは、適切にキャメルケースで命名されており、その名前も解答例に近いものであった。一方、有意率が最低のプログラムは、キャメルケースではなく、識別子に数値がついているなど、提案手法では対応していない形式であった。具体的には、`exName`、`usernumber`、`value2` などのような名前であった。

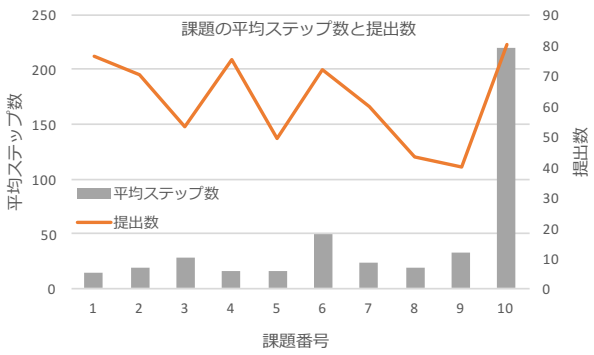

図 1: 課題ごとの平均ステップ数と提出数

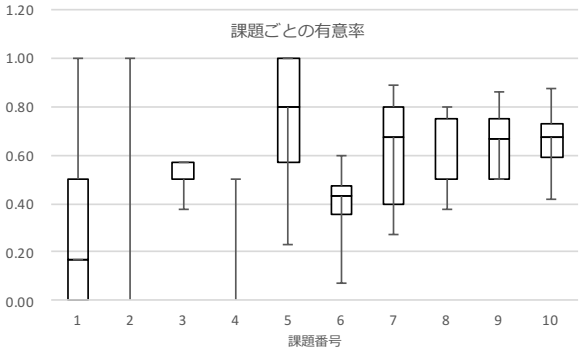

図 2: 課題ごとの有意率

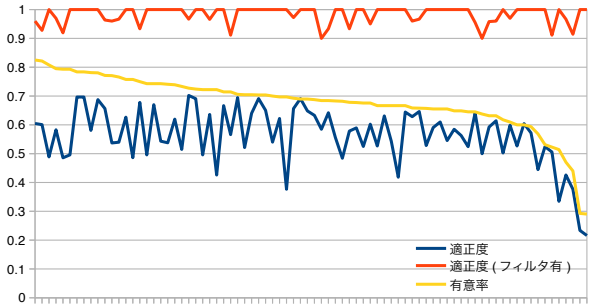

図 3: 課題ごとの有意率，適正度

3.2.4 識別子の良し悪しの判定

図 3 から今回は $M \geq 0.7 \wedge S \geq 0.6$ ならば良いプログラムとする。図 3 の中央付近のプログラムがどのような識別子名であるかを調査するため、サンプルとなる学生を 1 名選出し、詳細を調べた。そのプログラムのメトリクスは $M = 0.65, S = 0.54$ であった。実際にソースコードを確認すると、辞書にある単語を使っているものの、単語の繋ぎ方が不適切である為に、本手法での測定では低い値になったと考えられる。

4 まとめ

本稿では、初学者がつけた識別子名が適切か評価するために 2 つのメトリクスの提案を行った。提案したメトリクスで分析した結果、識別子に意味のある単語を用いた命名は初学者の大体は、できているが他人からだと動作などの予測が困難で製作者にしか分からない命名をしている事が分かった。分析結果から各メトリクスの閾値を決め、実際にプログラムの評価を行った。

参考文献

[1] WordNet-Princeton University cognitive science laboratory. `http://wordnet.princeton.edu/`, Last Access: 2016/1/7.