

大学のプログラミング教育における可読性の影響の調査

松本 翔太[†] 松原 俊一^{††} Martin J. Dürst^{††}

[†] 青山学院大学大学院理工学研究科理工学専攻知能情報コース

^{††} 青山学院大学理工学部情報テクノロジー学科

1 はじめに

大学におけるプログラミングの教育現場では、可動性が重要視されるあまり、可読性が軽視されがちな傾向が見られる。さらに可読性は明確な評価基準を設定しにくい要素であるため、教育現場において可読性が重要視されることはさらに難しくなる。しかし、プログラムを見たときにその構造や機能が容易に理解できることは作業効率の面から見ても重要だと考えられる。可読性の高いプログラム作りを心がけることで、より効率的な学習となり可動性にも良い影響を与える可能性がある。したがって、プログラミング教育においてその可読性の持つ影響力を詳しく調査することには意義がある。

筆者らの所属研究室はこれまで、学生が作成したプログラムの可読性に関する評価および可読性に対する意識の向上を目的とした研究を行ってきた。宮島ら [1] は、学生の記述スタイルと実際のプログラミング現場で使われている記述スタイルの間に隔たりがあると考えて、その差異について統計分析を試みた。どのような記述スタイルがよく使われているかというデータは本研究の指標にもなっている。さらに、松本ら [2] は、プログラムにおけるインデントの統一性を評価するシステムを構築した。それにより可読性を評価可能な項目にして、学生の可読性への意識を向上させようとした。

本研究は学生が作成したプログラムの内容を解析するために、このシステムを利用する。それによって得たデータを検証・解析して、可読性が大学のプログラミング実習においてどのような影響を持つのかを考察する。そしてプログラミング教育における可読性の影響力を明確にすることを目的としている。

2 統計解析

本研究では検定法という、ある仮説が正しいと比べてよいかどうかを統計学的・確率論的に判断するための

手法を利用する。その中でも特に母平均の差に関する検定が重要である。母集団が正規分布に従うとき、その平均の検定や2組の標本の平均に有意差があるかどうかの検定には t 検定がよく利用される。 t 検定とは、帰無仮説を立ててそれが正しいときに t 分布に従うという統計学的検定の一つである。 t 分布が直接もとの平均や標準偏差にはよらずに、自由度のみによることを利用している。

3 関連研究

Buseら [3] は、120人の被験者が10種類のオープンソースプロジェクトのソースコードの一部を評価して、読みやすさを採点している。得られた読みやすさのスコアと、25種類の要素との相関を求めている。要素にはコメント行数や予約語の数、空行の数、括弧の数、変数名の長さ、変数の個数などが含まれる。

この研究では読みやすさを直接評価してもらうことで数値化して、可読性に関わる要素との関係を調査している。対して本研究は、可読性の高さが可動性(学習の習熟度)に与える影響に注目した。プログラミング実習の課題を解析対象とすることでその得点を利用して、可読性に関わる要素との関係を調査する。

またBuseらの研究で使用されている要素は機械的に容易に抽出されるものに限られている。対して本研究は、可読性に関する要素の抽出をシステムによる自動的な方法だけでなく手動的方法でも行った。それにより人以外には理解の難しい感覚的な要素も調査対象にしている。

4 実験

4.1 データの抽出

本研究は、大学のプログラミング教育における可読性の影響を調査するために、学生が作成したプログラムにおける可読性と可動性の関係を解析する。そのために計算機実習Iという、本学科2年の学生が毎年100人ほど受講しているプログラミング実習を利用した。C言語を使用する実習の課題から2年分・7000個以上のプログラムを集めた。これらは全て実行テストで課題通りに動作することが確認されているものである。そしてそのプログラムから、自動的な要素の抽出と手作業による可読性に関する要素の抽出を行った。

Investigation about the Influence of Readability in University Programming Education

Shota Matsumoto[†] Shun-ichi Matsubara^{††} Martin J. Dürst^{††}

[†] Course of Information and Knowledge Engineering, Graduate School of Science and Engineering, Aoyama Gakuin University 5-10-1 Fuchinobe, Chuo-ku, Sagami-hara, Kanagawa 252-5258, Japan, ^{††} Department of Integrated Information Technology, College of Science and Engineering, Aoyama Gakuin University

c5610154@aoyama.jp, {matsubara, duerst}@it.aoyama.ac.jp

4.2 データの詳細

抽出した要素は 20 以上あり、学生の習熟度や可読性に関したものである。大まかな区別と説明は次のとおりである。

4.2.1 自動的に抽出した要素

まず可読性に関する要素の中で最も基本的な技法が字下げ（インデント）である。空白を用いてプログラムの構造を分かりやすくするために用いられ、特に条件分岐や繰り返し処理などの制御構造の範囲を明示させるために使用される。その使い方には多くのスタイルが存在し、特に規定されたスタイルは無い。しかし、同じプログラム、もしくは同じプロジェクト内ではスタイルを統一することが望ましい。

行数・バイト数はプログラムの大きさを示す指標である。例えば、動作が単純なプログラムに過大な量の記述が行われているのは非効率的だと判断できる。

学習の習熟度（可動性）を測る指標として様々な要素を利用する。例えば課題の提出時間である。決められた時間内に指示通りに動作するプログラムを提出できれば 100 点が与えられて、時間が遅れると点数が引かれるというやり方である。さらに課題の難易度によって重み付けも行っている。その他にも実習内で行われた試験の成績や、それらを総合的に採点したのものも利用する。

4.2.2 手動で抽出した要素

関数名・変数名はその頻出度から可読性の観点からも重要な位置を占めている。その名前は何を示している変数・関数なのか容易に理解できるような名前が望ましい。1 文字だけの場合や意味のない羅列の場合は、その役割を把握できずに動作を想像しにくくなる。

次に空行やトークンの間に入れる空白（スペース）である。プログラム中の様々な場所でスペースや空行を入れることにより、プログラムの中に余裕を持たせたり、つながりをわかりやすくすることができる。空行を挟み込むことで構造の大枠を把握して、動作の流れをつかむことができる。

プログラムの動作を明確に記述するために最も簡単な方法がコメントをつけることである。プログラムの要所で何をしているのか、何のためにしているのか、明瞭で自由な言葉を使ってコメントを残すことでプログラムの可読性は大いに上昇する。

これらの要素を利用して可読性と可動性との関係を解析する。

4.3 検定によるデータの解析

4.3.1 解析内容

収集したプログラムのデータは学生ごとにまとめて、可読性に関するデータをプログラムあたりの平均に変換する。その上で学習の習熟度に関する様々な要素の多寡によって学生のグループ分けを行う。そして優秀な集団とそうでない集団に分けて、その間にある可読性に関する要素の差を検定する。その結果から優秀な

集団に見られる特徴を抽出すれば、それが可読性に関する要素と学習の習熟度の関係性を導き出せる。

4.3.2 検定結果

優秀なグループにはいくつかの要素が多くなる傾向があった。その要素は、英語の変数名、過大な空白、総ライン数とバイト数、最長ラインの長さであった。インデントやコメントは単純な課題の得点で分類した場合は関係性が見られなかったが、講義内で行われた試験の成績を含めた総合的な成績を基準にすると深い関係があった。

5 考察

過大な空白や合計ライン数などの要素に関する結果は、やや過剰と判断できるほどに大きく広いプログラム作りが、構造の理解には有効だということと考えられる。課題の提出という時点ではインデントを重要視しなくてもよい結果を出すことはできる。しかしそのような学生は能力を測る試験の段階では優秀とはいえずなくなる。学習の習熟度と言う点で、インデントを気にかけることの意義が存在すると考察できる。

6 まとめ

本研究では大学のプログラミング実習における可読性の影響力を調査した。可読性と学習の習熟度との関係を解析して、習熟度が高い学生のプログラミングに見られる特徴を抽出した。今後も情報の収集・解析・考察のあらゆる面で、より効率的に有意なデータを集めていく予定である。データ収集の段階から細かく設定を練って、より多種多様な統計解析を行っていけば、因果関係まで含めた詳細なデータを得ることができる。そうして得られたデータをプログラミング教育の方法に活かしていくことが将来的な目標である。

謝辞

本研究は文部科学省科学研究費補助金（基盤 C，課題番号 21500905）の交付を受けている。

参考文献

- [1] 宮島明寛, 松本章代, Martin J. Dürst. オープンソースを用いた C 言語の記述スタイルの統計分析の試み. 情報処理学会第 71 回全国大会, 2009.
- [2] 松本翔太, 松本章代, Martin J. Dürst. C 言語用のプログラミングスタイル評価システムの構築. 情報処理学会第 72 回全国大会, 2010.
- [3] Raymond P.L. Buse, Westley R. Weimer. Learning a metric for code readability. *IEEE Transaction on Software Engineering*, Vol. 36, No. 4, pp. 546–558, 2010.