R-3

プログラムデバッグに基づくプログラミング学習手法の提案

橋浦研究室　　1165208 上村 勇太

1. 研究背景

ソフトウェア開発者の最も基本的なスキルはプログラミングであり，大学等の高等教育機関において様々な教育が行なわれている．

Ahmadzadeh ら[1]は，学生のデバッグスキルの習得度が高いほどプログラミング能力は向上するが，その逆は当てはまらないことを指摘し，これは一部の学生が他の学生よりもプログラムの実装をよく理解しているために発生すると述べている．このことから，学習者のこのようなスキルを集中的にトレーニングすることで，プログラミング能力の向上を促すことができるのではないかと考えた．

2. 研究目的

本研究は，プログラミングにおけるデバッグ過程に着目したプログラミング学習手法を提案する．

3. 提案手法

プログラムのデバッグ作業を行なうためには，バグのあるソースコードを用意する必要がある．一方，学習者がプログラミングに習熟するにつれて，自らが作成するプログラム中に含まれるバグは漸減するため，学習が進むほどプログラムをデバッグする機会は減少する．このため，学習者がデバッグ作業のみを集中的にトレーニングすることは難しい．

このような問題を解決し，デバッグ作業を好きなだけ行えるよう，ミューテーションテストツールを用いて学習者のプログラムに人為的かつ自動的にバグを埋め込む．これにより，学習者は常にバグが入ったプログラムが得られる．

4. 実現手法

環境の実装はミューテーションテストツール Mull[2]を用いて行なった．実装したシステムのイメージを図 1 に示す．

Mull によって注入される欠陥は演算子の置換，リテラルの書き換え等がある．

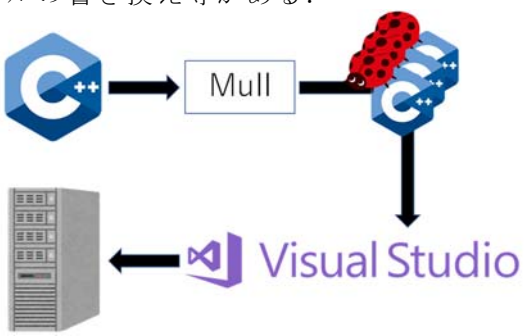

図 1 支援システムのイメージ

使用された問題を特定するため，ソースコードの 1 行目に UUID を付加して配信した．

5. 実験

以下の仮説を検証するために実験を行った．

- デバッグのトレーニングを行う前後でプログラミングの成績には有意な関係が存在する．

日本工業大学工学部情報工学科に所属する 8 名を被験者とする実験を行った．

各問題の終了条件は個人によるトレーニングを想定し，被験者が要件を満たしたと判断した時点とした．

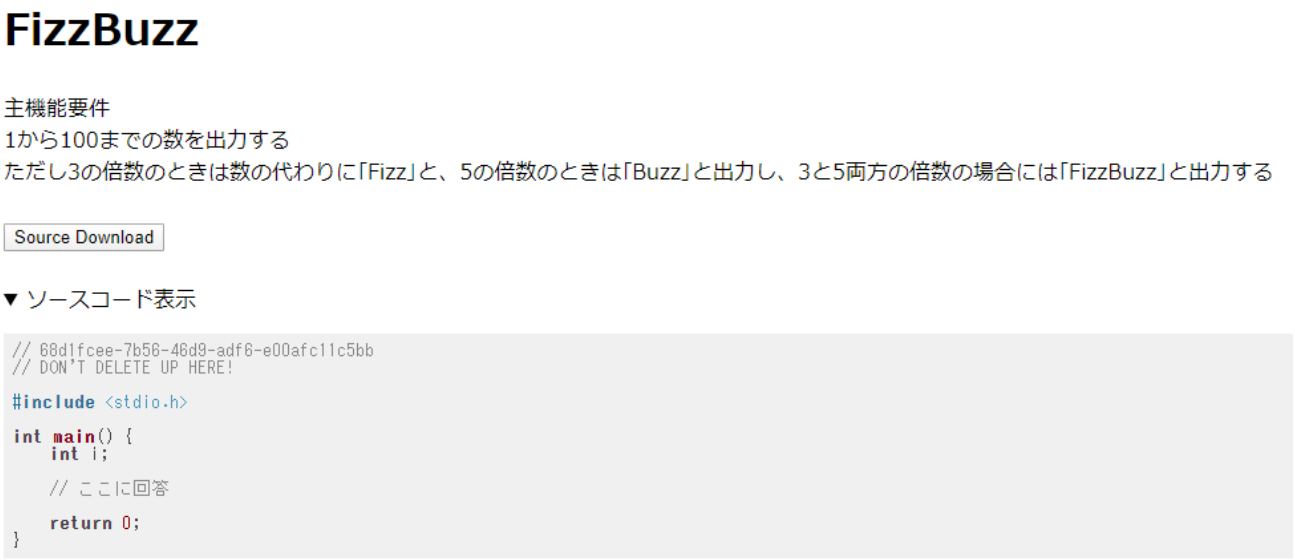

図 2 問題配信サイト

また，実験は以下の手順で行なった．

1. 事前テスト(1 時間)
   ① あらかじめ用意された要件定義に基づくコーディング課題 3 問
   ② 図 2 の問題配信サイトからソースコードをダウンロード
   ③ 要件を満たせるまで実装を行う
2. デバッグトレーニング(3 時間)
   ① 提案環境を用いたデバッグトレーニング 15 問
   ② 図 2 の問題配信サイトから生成された欠陥入りソースコードをダウンロード
   ③ 欠陥を取り除けるまで修正を行う
3. 事後テスト(1 時間)
   ① あらかじめ用意された要件定義に基づくコーディング課題 3 問
   ② 図 2 の問題配信サイトからソースコードをダウンロード
   ③ 要件を満たせるまで実装を行う

提出は個人を自動的に特定できるよう，VisualStudio の拡張機能として実装した．実装したツールのイメージを図 3 に示す．

図 3 提出ボタンの拡張機能

6. 評価方法

トレーニング前後のテストの成績の平均値に差があるのかを t 検定を用いて分析する．

テストの評価方法については，あらかじめ用意された要件定義をどれだけ満たせているかのチェックリストと，標準入出力がどれだけ一致しているかどうかで判定する．

入力で出力が変化する場合，入力に対応付けされた出力と比較する．

7. 結果と考察

本研究の提案手法が被験者の集団に影響を与えたかどうかを調査するため，2 つのテスト結果に対して，対応のある t 検定を行った．帰無仮説H_0を「ツールによる得点率の差に有意差はない」とし，対立仮説H_1を「ツールによる得点率の差に有意差がある」として検定を行った結果，表 1 の P 値は 0.05 より大きいとなり，対立仮説H_1が棄却され帰無仮説H_0「ツールによる得点率の差に有意差はない」が採択される．そのため，本手法をプログラミング学習に使用しても全体で見た能力は変わらないという結果になった．

事前テストと事後テストにおける得点率の被験者別で見た遷移を図 4 に示す．これにより，プログラミング能力が大きく向上するもの（3 人）と，ほとんど変わらないもの（5 人）が見られ，本環境が一部の学習者に対しては有効に作用していることが分かった．

表 1 ツールを用いた学習による t 検定の結果

	事後テスト	事前テスト
平均	0.818452381	0.741071429
分散	0.023596939	0.005537334
観測数	8	8
ピアソン相関	-0.236453889	
仮説平均との差異	0	
自由度	7	
t	1.177654394	
P(T<=t) 片側	0.138708592	
t 境界値 片側	1.894578605	
P(T<=t) 両側	0.277417183	
t 境界値 両側	2.364624252	

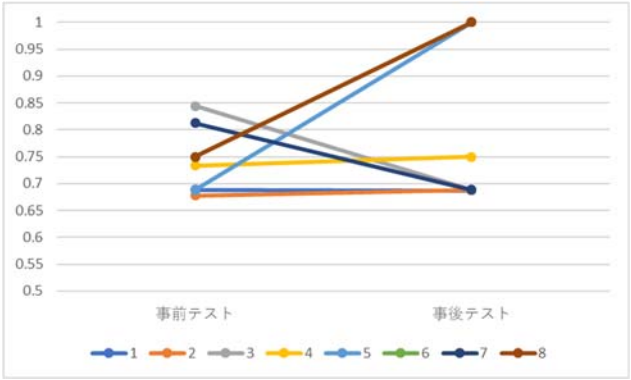

図 4 事前テストと事後テストの得点率の比較

8. まとめと今後の課題

本研究はデバッグスキルのトレーニングを行うことによるプログラミング能力の支援手法を提案し，実際に利用できる環境を構築した．実験により，提案環境がプログラミング学習に役立つ可能性が示唆された．

文　献

[1] Marzieh Ahmadzadeh, Dave Elliman, Colin Higgins, “The Impact of Improving Debugging Skill on Programming Ability,” Innovation in Teaching and Learning in Information and Computer Sciences, Vol.6, Issue 4, pp.72-87, Dec. 2015.

[2] Alex Denisov, Stanislav Pankevich, “Mull It Over: Mutation Testing Based on LLVM,” Proc. of 2018 IEEE Intl. Conf. on Software Testing, Verification and Validation Workshops (ICSTW2018), pp.27-31, Apr. 2018.