

Extension of Student Answer Analyzing Function to Five Problem Types in Desktop-version Java Programming Learning Assistant System

Ei Ei Mon[†] Su Sandy Wint[‡] Nobuo Funabiki[‡] Minoru Kuribayashi[‡]

1. Abstract

To enhance Java programming educations, we have developed the *Desktop-version Java Programming Learning Assistant System (D-JPLAS)*. *DJPLAS* supports *Element Fill-in-blank Problem (EFP)*, *Value Trace Problem (VTP)*, *Code Correction Problem (CRP)*, *Statement Fill-in-blank Problem (SFP)*, and *Code Writing Problem (CWP)*. Previously, we implemented the *student answer analyzing function* for EFP and CWP to help a teacher assessing performances of the students and giving feedbacks to them. In this paper, we extend the function to supporting all of the five problem types in *D-JPLAS*.

Keywords: JPLAS, Desktop-version, answer analyzing function, Java programming, student assessment

2. Introduction

Due to high reliability, portability, and scalability, the objected oriented programming language *Java* has been widely used to create various applications that can run on single computers or distributed system. The strong demands for Java programming education have appeared from IT industries. A typical programming course consists of grammar instructions in classes and programming exercises with computer operations.

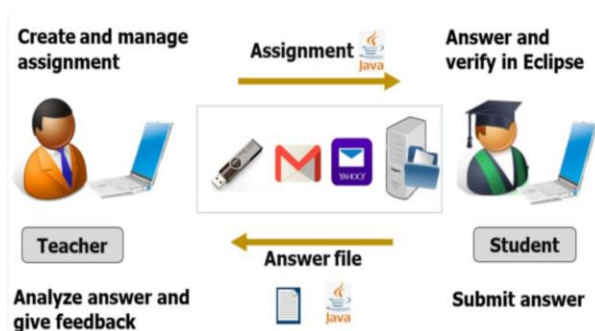


Figure1 Usage flow of D-JPLAS

To enhance Java Programming educations, we have developed the *Java Programming Learning Assistant System (JPLAS)* [1] as a Web application system, which can be used only in Internet available environments. Since

many students want to access *JPLAS* at no Internet-access places, we have implemented *Desktop-version JPLAS (D-JPLAS)*, which can run without the Internet [2]. Currently, *D-JPLAS* supports five types of programming problems that cover various stages of Java programming study.

Figure 1 illustrates the usage flow of *D-JPLAS*. The assignment files and answer test files should be shared between the teacher and the students using either USB memories, email, or file servers.

Currently, *D-JPLAS* offers *teacher service functions* for generating programming assignments, managing answer files from students, and analyzing their answers, and *student service functions* for solving and answering assigned problems. Unfortunately, in *teacher service functions*, the answers in EFP and CWP can be analyzed, to help a teacher assessing students' performances and giving feedbacks, by summarizing the answers that are submitted in several text files from students [3].

In this paper, we extend the function for analyzing answers to support all of the five problem types in *D-JPLAS*.

3. Extension of Student Answer Analyzing Function

In this section, we present the implementation of the extension.

3.1 Five Problem Types

The outlines of the five problem types are as follows:

- 1) Element Fill-in-blank Problem (EFP) requests to fill in the missing elements in the given problem code.
- 2) Value Trace Problem (VTP) requests to trace values of important variables.
- 3) Code Correction Problem (CRP) requests to fix the wrong source code according to the test code.
- 4) Statement Fill-in-blank Problem (SFP) requests to fill the statement given in the problem code.
- 5) Code Writing Problem (CWP) requests to write source code mentioned in the test code.

In EFP, VTP, and CRP, the answer consists of words whose correctness is marked by string matching with the correct answer. In SFP and CWP, the answer consists of statements whose correctness is marked by unit test on *JUnit* [4].

3.2 Input Data

As the input data to the student answer analyzing function, each row of the answer file contains the student ID, the problem ID, the answer date and time, and the answers and their marking results at each submission.

[†] eieimon@s.okayama-u.ac.jp

[‡] funabiki@okayama-u.ac.jp

Graduate School of Natural Science and Technology,
Okayama University, Okayama, Japan

- EFP, VTP, CRP: For example, the following data is described in the answer file:
32577781 2019-04-02 07:05:50
class[o],Car[o],num[o],gas[o],class[o],void[o],String
[o,)]o],Car[o],new[o],Car[o],new[o],Car[o],[x],
boolean[o],bl1[o], System[o],bl2[o]
Here, 32577781 describes the student ID,
2019-04-02 07:05:50 does answer date and time, class
does the answer by the student for the first question,
and [o] indicates the correct answer where [x] does the
incorrect answer.
- SFP, CWP: For example, the following data is
described in the answer file:
09428349 2019/04/24 13:07:05 p2.BinSortTest 5 1 0
Here, 09428349 describes the student ID, 2019/04/24
13:07:05 does answer date and time, p2.BinSortTest
does the package and class name, and 5 1 0 indicates
one test was failed among five tests.

3.3 Output Data

As the output data, the function outputs the average correct answer rate of each student for each problem.

- EFP, VTP, CRP:
 - 1) Count the total number of blank questions (=A) and the total number of correct answers (=B) of each student for each problem.
 - 2) Calculate the average correct answer rate by dividing B by A.
- SFP, CWP:
 - 1) Count the total number of tests in the test code (=C) and the total number of successful tests (=D) of each student for each problem.
 - 2) Calculate the average correct answer rate by dividing D by C.

No.	Student ID	Question id	Total Blanks	Correct Answers	Submissions	Average Rate
1	21411122	217	16	16	2	100%
2	21411122	219	8	8	2	100%
3	21411122	221	20	20	2	100%
4	21411122	223	16	16	2	100%
5	21411122	238	17	17	2	100%
6	21411122	242	17	17	4	100%
7	21411122	251	24	23	6	95.83%
8	21411122	253	8	8	6	100%
9	21411122	261	30	30	2	100%
10	21411122	265	18	17	4	94.44%
11	21411122	266	22	22	4	100%
12	21411122	281	23	23	2	100%

Figure 2 Answer result interface

3.4 User Interface

Figure 2 shows the answer result interface for a teacher, to help finding a student who cannot solve a problem well. This interface is implemented using Swing GUI [5]. The corresponding row is highlighted when the correct answer rate is smaller than the threshold that is given by the teacher.

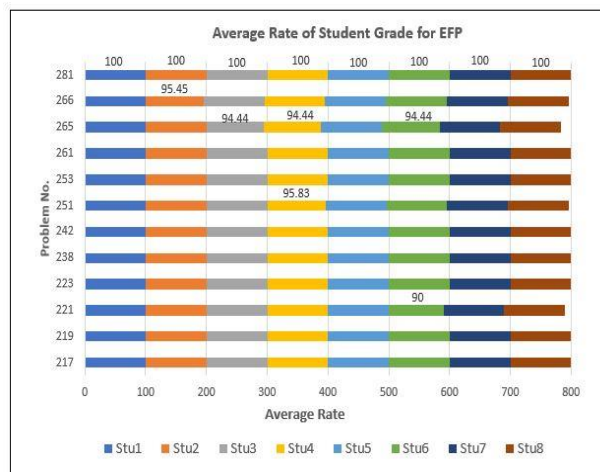


Figure 3 Answer rate graph interface

Figure 3 shows the average correct answer rates for the selected 12 EFP problems. For example, three students, *Stu3*, *Stu4*, and *Stu6* could not solve problem 265 fully.

4. Conclusion

This paper presented the extension of the student answer analyzing function to five problem types in D-JPLAS. In future works, D-JPLAS will be used and evaluated in Java programming courses in Japan and Myanmar.

References

- [1] S. Ao et al. edited, IAENG transactions on engineering sciences special issue for the international association of engineers' conferences 2016 (volume II), pp. 517-530, World Sci. Pub., 2018.
- [2] S. S. Wint, N. Funabiki, and M. Kuribayashi, "Design and implementation of desktop-version Java programming learning assistant system," in Proc. HISS, pp. 254-257, Nov. 2018.
- [3] E. E. Mon, S. S. Wint, N. Funabiki, and M. Kuribayashi, "Implementation of student answer analyzing function for desktop-version Java programming learning assistant system," in Proc. IEICE General Conf., pp.74-75, Mar. 2019.
- [4] JUnit, <http://www.junit.org/>.
- [5] Swing GUI, <https://docs.oracle.com/javase/tutorial/uiswing/index.html>.