Graphical Tutoring of Message Passing to Promote Understanding of Java Framework

Atsuko Ueno, Hiroshi Taguchi, and Hiromitsu Shimakawa

(Speaker: Hideyuki Takada)
Background (No.1)

- Difficulties in Learning Java Programming
  - Many Constraints in Programming
    - Java Framework
      - characteristics of each component and how to build the components
    - Message Passing
      - sending/receiving information among objects

/*set the layout */
container.setLayout
(new BorderLayout());

Who is the sender?
Who is the receiver?
What arguments are sent?
What the sender can refer?

Message Passing

Sender

main()

container

Receiver

setLayout

argument : new BorderLayout();
Programming Education of Most Universities

- Combination of lecture class and exercise class
  - In lecture classes, programming knowledge is taught.
  - In exercise classes, students try to write codes using the knowledge.
- Many of students cannot utilize the knowledge only in lecture classes
  - A huge gap between explanation in lectures and coding from scratch.
  - Some support is desirable between lectures and exercises.
Target and Assumption

- Target
  - Students who learn the syntax, but do not understand the Java framework and the message passing.

- Assumption
  - It is a self-learning tool for students to understand the Java framework and the message passing.
  - After each lecture class, they use our tool before an exercise class.
Proposed Tool

- It shows a program on a sequence chart.
- It checks the code dependency during programming.
- The tool gives advices during their programming.
- Students understand what is necessary for message passing.

Students understand what is necessary for message passing.
The tool gives advices during their programming.
Effective for students to understand
- how to use the Java framework, and
- the mechanism of message passing.
Two Important Items for Coding

The items make students understand what is necessary to state a Java program.
When we must describe code A after code B, we say “A depends on B”, and denote $A \rightarrow B$. 

Here is the code example:

```java
/* create a frame */
JFrame frame;
frame = new JFrame("Title");

/* get the pane in the frame */
Container container = frame.getContentPane();
```
Classification of Dependency Relationships

- Dependency relationships can be classified based on their causes
  - A message cannot be sent to an instance if the receiver instance has not been created.
    → Code Dependency coming from Instance Creation
  - The order of codes has been determined in the Java Framework.
    → Code Dependency Coming from Java Framework Rules.
Examples in GUI Programming

/* code 1: Button Creation*/
JButton button = new JButton();

/* code 2: Layout setting*/
container.setLayout(new BorderLayout());

/* code 3: Place the button in the center*/
container.add(button, BorderLayout.CENTER);

Code Dependency Coming from Instance Creation

Code Dependency Coming from Java Framework Rules
Mechanism of Tutoring on the Tool

- Our tool checks code dependency when a student specifies one code.
- The tool advises him if he cannot find anything the code depends.
- The advice is given from a personalized tutor in the tool.
Implementation of Tutoring Mechanism

- For each exercise, the following information is specified in a text file.
  - Instances and methods necessary to solve the exercise.
  - For each method, codes on which the method depends.
  - The cause of each dependency
Implementation of Tutoring Mechanism

Student Coding

- Creation of Instance A
- Creation of instance C
- A.method1();
- A.method2();

The student specifies method2 of A

It depends on the creation of instance B

B has not been created yet

Advice

A
- method
  - method1
  - method2

depends on creation of instance B

method

Cause of dependency
Experiment

- Small Lecture and Exercise Using Our Tool
  - 10 undergraduate/graduate students
    - 30 minute programming test to check their ability
    - 30 minute Java lecture
    - 30 minute self-learning
    - 10 minutes yes/no test to check their knowledge
    - 50 minute programming exercises(Ex.1, Ex2)

- A half of the students use the tool for the self-learning, while the remains use only textbook.
## Result

- **Score of the tests (The full mark is 100)**

<table>
<thead>
<tr>
<th>Programming</th>
<th>without tool</th>
<th>With tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary test</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td>Ex.1</td>
<td>87</td>
<td>91</td>
</tr>
<tr>
<td>Ex.2</td>
<td>65</td>
<td>73</td>
</tr>
<tr>
<td>Yes/no test to check knowledge</td>
<td>74</td>
<td>68</td>
</tr>
</tbody>
</table>

- **Questionnaire**

(5: Strongly agree, 4: Agree, 3: Neutral, 2: Be against, 1: Be strongly against)

<table>
<thead>
<tr>
<th>Statement</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>The tool contributes to understanding of Java programming</td>
<td>4</td>
</tr>
<tr>
<td>The tool contributes to understanding of Swing framework</td>
<td>3.6</td>
</tr>
<tr>
<td>Self-learning with the tool promotes understanding</td>
<td>3</td>
</tr>
<tr>
<td>The sequence chart is effective for understanding</td>
<td>3.8</td>
</tr>
<tr>
<td>Tutoring of the tool is useful to identify mistakes</td>
<td>3.8</td>
</tr>
</tbody>
</table>
Discussion

- Students with the tool have got good scores.
  - The tool is effective for Java programming education

- But their scores are poor in the knowledge test.
  - The tool gives advice only when they make mistakes.
  - Students are not conscious of their wrong understanding of the knowledge.

- Evaluation for self-learning with the tool is poor in the questionnaire.
  - The tool is still not easy to use.
  - The user interface of the tool should be improved.
Conclusion

- A tool to Support Understanding of Java Framework and Message Passing

- Sequence Chart and Code Dependency

Evaluation

- The tool is effective for Java programming education.
- The user interface should be improved.