INTRODUCTORY PROGRAMMING EDUCATION USING SCRATCH AND JAVA

Toshie Matsumura [*a], Yoshihiro Ichikawa [a] and Yoshimichi Honma [a]

[a] National Institute of Technology (KOSEN), Nara College, Department of Information Engineering, Yamato Koriyama City, Japan

*E-Mail namiki@info.nara-k.ac.jp

Abstract

In the National Institute of Technology (KOSEN), Nara College curriculum, we teach programming to students in grades 2-4. The unit of programming is 6 credits in total. Students learn the Java language for the first two years. Students learn C language in the last year. In other lectures, students may also use other languages, such as Python and C#, for experiments and exercises. In this way, in the curriculum of this department, we teach programming firmly, but it cannot be said that students have programming skills. As a cause of this, a certain number of students are not good at programming at the stage of the lower grades. We consider that students should submit assignments and pass exams instead of actively learning to program. Therefore, to dispel students' weak perception of programming, on the other hand, for students who have advanced understanding, we will change the tasks according to their proficiency level so that they can challenge more advanced problems. This is possible by having multiple instructors conduct classes, and there were problems such as difficulty in evaluating students by dividing them according to their proficiency level. Therefore, we gave a lecture using Scratch and the Java language in the Introduction to Computer Programming class held in the second grade, which is the first introductory education, and we report the details in this paper. In this class, the following procedures were carried out to help students to understand the control structures that they should learn first: Sequential processing, Conditional branching, and Iteration. (1) Visual programming (Scratch). (2) Java language programming. (3) Application problem. We teach classes with (1) to (3) as one set. This was not just a matter of teaching grammar and doing exercises, but to help students understand and maintain their motivation. We compared and considered the results of classes conducted using this method for the first four years from 2019 to 2022 with classes conducted using only Java. Although the time required to teach grammar content was shortened compared to the conventional class method that taught only Java, the grades did not deteriorate compared to before, and it is suitable for classes to learn Java language and C language from the second and third years. There were no major issues.

Keywords:
Programming, Introductory Education, Scratch, Java

1. Introduction

In 2020, programming became compulsory in elementary schools, and programming education started in elementary school. Among them, "Acquiring the ability to understand and use computers well is important for children living in the future society where computers are required to be used in all activities. It is extremely important even if I get a job.” [1] Reference materials have been published for this purpose [2]. It uses visual programming languages such as Scratch [3] and Viscuit [4]. At Nara College of Information Engineering, we have implemented programming education as part of our curriculum. We also found out that there are students who are not good at programming, other than students who are interested in programming. In response to this, we have taken measures such as setting tasks by proficiency level, but we have not seen any significant improvement.[5] Therefore, we considered seamlessly connecting logical thinking and programming languages and using visual programming, which is familiar in elementary schools, in the introductory programming education of Nara College of Information Technology Department. In this paper, I report the contents of the implementation and hope that it will be of help to future programming education in technical colleges.

2. Curriculum

We design specialized subject curricula. The curriculum consists of hardware, software, network, and information security fields.

Programming is positioned in the software field. Students learn programming in grades 2-4[6]. Table 1 shows the number of credits and grades for programming subjects. We teach programming introductory education in the Introduction to Computer Programming class.

Students learn the Java language for the first two years. Students learn C language in the last year. In class, students may also use other languages, such as Python and C#, for experiments and exercises. In this way, in the curriculum of this department, we teach programming firmly, but it cannot be said that students have programming skills. As a cause of this, a certain number of students are not good at programming at the stage of the lower grades. We consider that students should submit assignments and pass exams instead of actively learning to program. Therefore, to dispel students' weak perception of programming, on the other hand, for students who have advanced understanding, we will change the tasks according to their proficiency level so that they can challenge more advanced problems [6]. This is possible by having multiple instructors conduct classes, and there were problems such as difficulty in evaluating students by dividing them according to their proficiency level.

Therefore, we decided to conduct a lecture using Scratch and the Java language in the Introduction to Computer Programming class that was held in the second grade, which was the first introductory education. This paper reports the contents.

Table.1 Number of credits and Grades

Subject	grade	Number of credits
Introduction to Computer Programming	2nd year, first semester	1
Computer Programming I	2nd year, second semester	1
Computer Programming II	3rd year, full year	2
Computer Programming III	4th year, full year	2

3. Lecture Structure

In this lecture, in the 15th week of the semester, we implemented a procedure to understand the control structures that should be learned first: sequential, branching, and iteration.

(1) Visual programming (Scratch)
(2) Java language programming
(3) Application problem

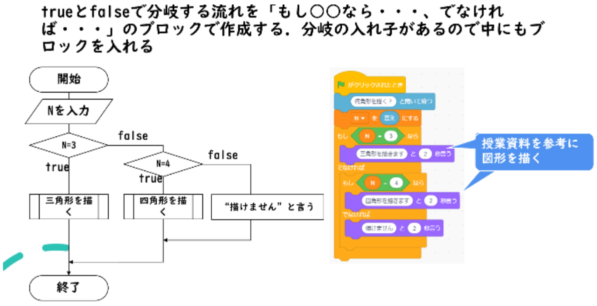
We teach classes with (1) to (3) as one set. This was not just a matter of teaching grammar and doing exercises, but to help students understand and maintain their motivation.

For example, Table 2 shows the contents of classes for the 2022 academic year. Except for guidance and comprehensive exercises, Scratch, Java, and application problems are given as lectures. Comprehensive questions are those in which students use what they have learned to solve problems. In addition, one part of the applied problems was studied in groups.

The lecture materials used were prepared by the instructor. Figure 1 shows an example of the lecture materials for the visual programming session. Fig. 2 shows an example of the lecture materials for programming using Java. The lecture materials were distributed to the students so that they could look back at them even after the project was completed. Scratch and Java lecture materials include flow charts and explanations to make them easier to understand visually.

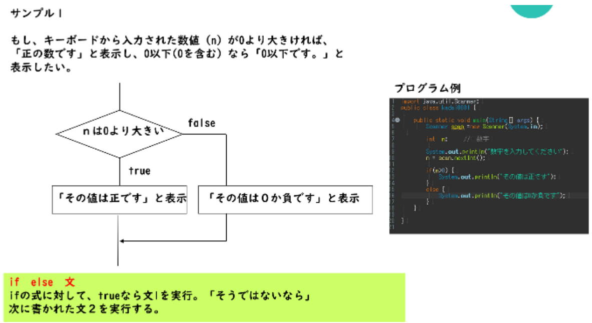
Table.2 Week and Lecture content (2022)

Week	Content
1	Guidance How to use scratch/How to use the Java development environment
2	Scratch: sequential processing
3	Java:　sequential processing
4	Application problem, how to debug
5	Scratch: conditional branch, flowchart
6	Java: conditional branch
7	Application problem
8	Midterm Exam
9	Scratch: iteration
10	Java: iteration
11	Application problem
12	Comprehensive question
13	Comprehensive question
14	Comprehensive question
15	Final Exam
16	Test return, confirmation


trueとfalseで分岐する流れを「もし○○なら・・・、でなければ・・・」のブロックで作成する．分岐の入れ子があるので中にもブロックを入れる

授業資料を参考に図形を描く

Fig.1 An example of a lecture material (Scratch)


サンプル１

もし、キーボードから入力された数値（n）が0より大きければ、「正の数です」と表示し、0以下(0を含む）なら「0以下です。」と表示したい。

プログラム例

```
import java.util.Scanner;
public class Kadai0001 {

	public static void main(String[] args) {
		Scanner scan = new Scanner(System.in);

		int n;	// 数字

		System.out.println("数字を入力してください");
		n = scan.nextInt();

		if(n>0) {
			System.out.println("その値は正です");
		}
		else {
			System.out.println("その値は0か負です");
		}
	}
}
```

if　else　文
ifの式に対して、trueなら文１を実行。「そうではないなら」次に書かれた文２を実行する。

Fig.2 An example of a lecture material (Java)

4. Results and Discussion

We compared the four-year course from 2019 to 2022 with the proposed method and the course from 2015 to 2018 taught students only the Java language. Since the Introduction to Computer Programming lecture is in the first semester, the lectures to be compared are also the same semester for second-year students. Therefore, we compared the results of the previous semester's classes.

Both lectures taught control structures, and they taught grammatical items excluding the Class structure.

The proposed method's grade standard deviation was 11.05-14.31 and the median was 77.0-95.8. On the other hand, in the conventional method, the grade standard deviation was 11.27-20.20 and the median was 70.9-79.5.

Although the teaching time for grammatical content was shortened compared to the conventional teaching method that teaches only Java, grades did not deteriorate sharply compared to before 2018. The median is rising. We considered this cause.

There are differences between full-year lectures and half-year courses, as well as different lecture contents and instructors, so a simple comparison cannot be made. A high median means that students performed well overall, even though the standard deviations do not differ significantly. We attribute this to the overall high performance of assignment submissions and exams. We believe that the easy-to-understand lectures and easy assignments for the students did not make them feel that they were not good at programming. Although we only teach the bare minimum of grammatical items, I thought that a virtuous cycle has been created in which students were able to research and learn on their own. I haven't formally taken a survey of students, but when I hear from students, they don't seem to have a strong sense of weakness. In addition, there is no problem in the following lectures on Programming I, Programming II, and III. Despite the short amount of time, it does not affect the grade.

4. Conclusions

Although the teaching time for grammar content was shortened compared to the conventional teaching method that teaches only Java, the grades did not deteriorate compared to before, and the result was a high evaluation. This is not accurate due to differences in faculty, exam questions, and the comparison between semester and full-year courses. However, we believe that low evaluations will foster a sense of inadequacy and that it will not affect the learning of the Java and C languages in the latter half of the second year and beyond. Also, students need to understand the control structure of the program.

It will be some time before elementary school students who have programming as a compulsory subject enter technical colleges, but we hope that the practical programming education that we provide at our school will help other technical colleges.

Future issues include the following. We have not been able to investigate whether students are not good at programming using questionnaire surveys. It is necessary to verify how much the number of students who are not good at it can be reduced by using the proposed method. Also, the time spent teaching Java grammar is reduced, while students need time for self-study to solve problems. Since this verification has not been done, this is a future task.

References

[1] Guide to Elementary School Programming Education： Ministry of Education (mext.go.jp) Retrieved from https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm

[2] Training materials for elementary school programming education： Ministry of education (mext.go.jp) Retrieved from https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416408.htm

[3] Scratch - Imagine, Program, Share (mit.edu) Retrieved from https://scratch.mit.edu/

[4] viscuit computer is clay!! Retrieved from https://www.viscuit.com/

[5] Shinji Uchida, Toshie Matsumura, Takayuki Nishino, "Trial of programming education that introduces proficiency", Jouhou Kyouiku Kenkyuu Shuukai kouen rombunshuu, 120-123, 2008

[6] NIT, Nara college, curriculum Syllabus 2022 Retrieved from http://syllabus.kosen-k.go.jp/Pages/PublicSyllabus?school_id=28&department_id=15&subject_id=0031&year=2021&lang=ja