

A Case Study of Errors Occurring during Python Programming Education for Elementary Teachers

Junghee Jo*

*Associate Professor, Dept. of Computer Education, Busan National University of Education, Busan, Korea

[Abstract]

In this paper, various types of errors made by elementary school teachers during the process of learning Python were analyzed to enhance the effects of text-based programming education. This study found that the most common type of error made by elementary school teachers was logical errors (67%). This is thought to be because teachers need to organize the meaning of a given task systematically which demands higher levels of programming ability, rather than simply acquiring syntax. Prior experience with text-based programming languages was found to have a significant impact on learning Python, unlike block-based programming languages. These findings can be used as foundational data to develop educational curricula to teach elementary school teachers text-based programming languages.

▶ **Key words:** Python, Elementary School Teacher, Error, Education, Programming

[요 약]

본 논문은 초등교사들의 파이썬 기반의 데이터 분석 프로그래밍 학습 과정에서 발생하는 다양한 오류들을 분석하여 텍스트형 프로그래밍 교육의 효과적인 방법을 탐색하는 것을 목적으로 한다. 분석 결과, 초등교사들이 가장 많이 발생시킨 오류의 타입은 논리 오류(67%)였으며 단순한 문법의 습득보다는 주어진 과제의 의미를 체계적으로 정리하여 논리적으로 접근하여야 하므로 높은 난이도의 프로그래밍 역량이 요구되기 때문이라고 추측된다. 텍스트형 프로그래밍 언어를 사용하는 교사들의 선행학습은 블록형 언어와는 달리 파이썬의 학습에 유의미한 영향을 주는 것으로 나타났다. 본 연구의 결과는 향후 초등교사 대상의 텍스트형 프로그래밍 교육의 기초 자료로 활용될 수 있을 것이다.

▶ **주제어:** 파이썬, 초등교사, 오류, 교육, 프로그래밍

• First Author: Junghee Jo, Corresponding Author: Junghee Jo
*Junghee Jo (dreamer@bnue.ac.kr), Dept. of Computer Education, Busan National University of Education
• Received: 2024. 12. 13, Revised: 2024. 12. 30, Accepted: 2025. 01. 07.

I. Introduction

인공지능 시대가 도래하면서 프로그래밍은 전문가 고유의 영역에서 벗어나 누구나 학습하여야 하는 필수 소양으로 새로이 인식되고 있다. 이러한 변화를 반영하여 2022년 개정 교육 과정에서는 미래 사회가 요구하는 역량의 함양 및 디지털 사회에서 발생하는 문제를 해결하기 위한 중요한 능력으로 '프로그래밍'을 강조하고 있다[1].

국내 초등 교육 현장에서는 스크래치 또는 엔트리와 같은 블록형 언어를 활용한 프로그래밍 교육 기반의 연구가 다양한 측면에서 수행되고 있다[2-4]. 최근에는 블록형 언어뿐만 아니라 텍스트형 프로그래밍 언어에 대한 습득도 부가적으로 요구되기 시작하였으며 이에 따라 현직 초등 교사들이 갖추어야 하는 프로그래밍 역량 수준도 높아지고 있다[5-7]. 파이썬은 대표적인 텍스트형 프로그래밍 언어로서 교육을 목적으로 개발된 것은 아니지만 문법이 간결하여 직관적이며 접근성이 좋으므로 초보자가 학습하기에 적합한 구조이다[8].

이러한 프로그래밍 언어를 이용한 소프트웨어 개발 과정에서는 필연적으로 오류가 발생할 수 있는데 경험이 부족한 초보자의 경우는 전문가의 도움 없이 스스로 해결하기 어려운 부분이 있으므로 프로그래밍 학습에서 매우 중요한 문제로 인지되고 있다. 이에, 초보 학습자가 발생시키는 오류 유형을 분석하고 해결하는 방법과 관련된 내용이 활발히 연구되고 있다[9-12].

본 연구에서는 현직 초등교사들을 대상으로 오픈 데이터 분석을 목적으로 교과 시간에 파이썬 프로그래밍을 학습시키는 과정에서 교사들이 발생시키는 다양한 오류들을 수집하였다. 그리고 수집한 데이터를 분석하여 각종 오류를 유형별로 분류하고 이러한 오류를 발생시키는 원인을 파악하여 향후 교육에서 유사한 오류를 줄이는 방안을 도출하였다. 또한, 교사들의 블록형 또는 텍스트형 프로그래밍 언어의 선행학습이 파이썬 언어의 학습에 긍정적인 영향이 있는지 분석하였다.

II. Related Works

1. Definition of programming errors

프로그래밍 오류는 프로그램 실행 중에 런타임 오류를 발생시킬 수 있는 코드를 의미한다[13]. 예를 들어, 문자열에 따옴표를 빠뜨리거나 괄호를 빠뜨린 조건식 등이 포함된다. 또한, 프로그램의 출력이 문서화된 요구 사항 또는

프로그래머의 원래 의도와 일치하지 않을 때도 발생할 수 있다[14].

2. Types of programming errors

프로그래밍 오류는 하나의 체계로 단순하게 분류하기는 어려우며 Table 1과 같이 연구자들의 관점에 따라 다양한 방법으로 분류되었다.

Table 1. Prior Studies Classifying Programming Errors

Study	Classification
Moon[9]	Logical errors, Operation errors, Data errors, Interface errors, Other errors
Name[10]	Syntax errors, Logical errors, Coding errors, Robot operation errors
Kim[11]	Syntax errors, Logical errors, Coding errors
Jung[12]	Logical errors, Syntax errors, Name errors, EOL errors, EOF errors, Type errors, Attribute errors, Value errors, Turtle graphics errors, Index errors, Indentation errors

문외식(2018)은 예비교사를 대상으로 실시한 파이썬 교육 과정에서 수집한 오류 데이터를 논리, 연산, 데이터, 인터페이스, 기타 오류의 5가지 항목으로 분류하였다[9]. 남재원(2009)은 초등학교 정보 영재 학생들을 대상으로 실시한 NXC 언어 기반의 로봇 프로그래밍 교육 과정에서 수집한 오류를 문법, 논리, 코딩, 로봇 동작 오류로 분류하였다[10]. 김지선(2014)은 대학 영재교육센터에서 C언어 강좌를 수강한 중고등학생 대상의 온라인 교육에서 수집한 오류 데이터를 남재원(2009)이 분류한 오류 유형을 기반으로 문법, 논리, 코딩오류의 3가지 항목으로 분류하였다[11]. 정혜욱(2021)은 대학 교양과목의 수강자들을 대상으로 파이썬 교육을 실시하여 수집한 오류 데이터를 코드의 실행과정에서 출력되는 오류 메시지와 파이썬 홈페이지 문서의 오류 내용을 참고하여 Table 1과 같이 11가지로 분류하였다[12].

III. Research Design

1. Research Subject

본 연구는 교원양성대학의 대학원에 재학 중인 현직 초등교사 21명을 대상으로 진행되었다. 본격적인 수업을 시작하기 전에 본 연구의 참고 자료로 활용하기 위하여 프로그래밍 경험에 관한 설문을 시행하였다. 설문지 내용은 경험이 있는 프로그래밍 언어의 명칭, 능숙도, 학습경로, 그리고 사용기간으로 구성하였다.

2. Research Procedure

학생들은 15주에 걸쳐서 파이썬을 교육받았으며 세부적인 강의 내용은 Table 2와 같다. 강의는 파이썬 전반에 걸친 모든 문법의 내용을 다루는 것이 아니라 기초적인 데이터 분석을 위한 최소한의 문법 내용과 오픈 데이터를 활용한 데이터 시각화 위주의 간단한 분석, 그리고 전반적인 실습에 대한 리뷰로 실시되었다.

Table 2. The Curriculum Used for the Lecture

Category	Contents
Survey	Programming background
Basics	Overview of data science
	Overview of Python
	Overview of grammar in Python
	Overview of Colab
Data analysis	Overview of Matplotlib
	Visualization using weather data
	Visualization using public transportation data
Review	Visualization using population data
	Text encoding
	Diversity in programming
	Importance of indentation
	Understanding of file pointer
	Using variables with the same name as Python built-in functions
Final evaluation	Shortcuts for comments
	Examination using programming

Basics 영역은 이론과 실습을 병행하여 강의를 이루어 지지만, Data analysis 영역에서는 주로 실습을 위주로 강의를 진행되며 학생들에게 문제를 배포하고 Basics 영역에서 습득한 지식을 기반으로 제한된 시간 내에 각자 해결하도록 하고 학생들이 작성한 파이썬 코드를 점검하면서 발생한 오류들을 수집하였다. 강의 후반에는 Review 시간을 할당하여 수집된 오류들을 기반으로 학생들이 코딩 과정에서 흔히 발생시키는 오류와 해결하지 못하는 오류들을 정리하여 설명하였다.

15주 차에는 평가의 하나로 프로그래밍 시험을 실시하여 제한된 시간 내에 학생들이 각자 작성한 소스 코드를 제출하도록 하였다. 시험 내용은 파이썬 코딩을 통해 교수자가 배포한 CSV 파일로부터 데이터를 읽어와서 그래프를 생성하고 간단한 분석 결과를 화면에 출력하는 내용으로 구성하였다. 학생들이 텍스트 프로그래밍의 초보임을 고려하여 난이도는 수업 중에 실시한 실습 내용에서 크게 벗어나지 않도록 하였다. 기말 평가에서 제출한 파이썬 코드에서 발견한 오류들도 교수자가 강의중에 수집한 오류들과 함께 본 연구에 활용하기 위해 수집하였다.

IV. Results

1. Classification of Error Types

본 연구에서 수집된 오류를 분석하여 분류한 결과는 Table 3과 같다. 선행 연구(9-12)를 기반으로 본 연구에서는 초등교사들이 발생시킨 오류들을 논리 오류(Logical Errors), 문법 오류(Syntax Errors), 데이터 오류(Data Errors), 코딩 오류(Coding Errors)의 4가지 타입으로 분류하였다. 그리고, 각 타입별로 학생들이 자주 발생시키는 오류들을 세부 항목으로 분류하고 식별을 위해 코드 항목을 생성하였다.

Table 3. Type, Code, and Description of Errors

Type	Code	Description
Logical Errors	LE_1	Incorrect indentation
	LE_2	Declaring variable initialization within a loop
	LE_3	Missing some of requirements in implementation
	LE_4	Incorrect use of a string that looks like a number
	LE_5	Lack of understanding of the data structure including List
	LE_6	Lack of understanding of the usage of a function or keyword
	LE_7	Failure to understand the behavior of the file pointer in a nested loop
Syntax Errors	SE_1	Errors due to attempting to concatenate strings and variables using + operator
	SE_2	Error caused by not being able to distinguish between a string that looks like a number and a number
	SE_3	Errors due to using == operator instead of = operator to store value in variable
	SE_4	Errors due to using = operator instead of == operator to compare two values for equality
	SE_5	Errors due to attempting to declare and use a variable with the same name as a Python built-in function
	SE_6	Other errors due to grammatical inconsistency
Data Errors	DE_1	Errors occurring when converting data in Excel format to CSV format
	DE_2	Errors occurring when attempting to process data files that contain garbage values
	DE_3	Errors when using a non-existent file name
Coding Errors	CE_1	Errors caused by missing : or misspelling as ; in loop or conditional statement
	CE_2	Error due to typo in function name or variable name
	CE_3	Errors caused by missing ' ' or " "
	CE_4	Errors caused by missing (or)

논리 오류는 파이썬에서 오류 내용을 명시적으로 출력하지는 않으나 주어진 과제를 논리적으로 접근하지 못하여 의도하지 않은 실행 결과를 생성하는 오류이다. 문법 오류는 파이썬에 관한 문법적인 지식이 부족하여 발생시키는 오류이다. 함수에서 불필요한 매개변수를 사용하거나 빠뜨리는 경우도 이에 해당하며 논리 오류와는 달리 파이썬에서 프로그래머에게 오류 메시지를 명시적으로 제공한다. 데이터 오류는 데이터를 핸들링하는 과정에서 발생시키는 오류로써 데이터의 생성, 정제, 또는 데이터를 읽어오는 과정에서 발생한다. 마지막으로, 코딩오류는 프로그래밍 과정에서 괄호의 누락과 같이 단순 오타로 인하여 발생하는 오류이다.

2. Description of the Error

파이썬의 학습 과정에서 초등교사들에 의해 발생한 오류를 분류한 Table 3에 명시된 각 코드 항목에 관한 상세한 내용과 오류를 발생시키는 원인을 설명하기 위한 예제는 다음과 같다.

- 들여쓰기 (LE_1)

대부분의 텍스트 프로그래밍 언어는 들여쓰기가 코드의 실행 결과에 영향을 주지 않도록 설계되어 있지만, 파이썬은 들여쓰기에 민감하므로 다양한 오류를 생성하였다.

- 변수 초기화 구문의 위치 (LE_2)

변수를 초기화하는 코드를 반복문 내에 작성하여서 변수에 저장된 값이 유지되지 않고 계속하여 초기화되는 현상이 발생하였다.

- 일부 기능의 구현 누락 (LE_3)

프로그램의 요구 사항 중에 일부 기능의 구현이 누락되었다.

- 문자열 형식의 숫자 (LE_4)

문자열 형식의 숫자를 int() 또는 float() 함수를 사용하여 숫자형으로 명시적으로 변환하지 않고 그래프를 생성하고자 시도하여 의도하지 않은 실행 결과가 발생하였다.

```
for row in data :
    male = row[1:4]
    plt.plot(male)
    plt.show()
```

Fig. 1. Example of LE_4, incorrect use of a string that looks like a number

예를 들어, Fig. 1에서 row 변수에 ['6266', '7136', '7634']의 문자열 형식의 숫자들이 저장되어 있다고 가정할 때 이들을 숫자형으로 변환하지 않고 plot 함수에서 사용하면 문법적 오류는 발생하지 않지만 파이썬이 숫자를 인지하지 못함으로써 의미없는 그래프가 생성되었다.

- 데이터 구조 (LE_5)

데이터 구조, 특히 List의 사용법을 정확히 이해하지 못하여 오용하는 사례가 자주 발생하였다. 예를 들어, Fig. 2의 myList에서 'a'를 가져와서 출력하고자 의도하였으나 첫 번째 print 구문이 아니라 두 번째 print 구문을 사용함으로써 'a'가 저장된 List인 ['a']를 가져오는 경우와 같이 의도하지 않은 실행 결과가 발생하였다.

```
myList = ['a', 'b', 'c']
print(myList[2])
print(myList[2:3])
```

Fig. 2. Example of LE_5, lack of understanding of the data structure including List

- 함수 또는 키워드 (LE_6)

함수 또는 키워드의 사용법을 정확히 인지하지 못하여 오용하는 사례가 발생하였다. 예를 들어, range(5)의 의미는 0부터 순차적으로 5개의 값을 생성하는 것인데 1부터 생성하는 것으로 잘못 판단하거나 break 키워드를 오용하여 의도하지 않은 실행 결과가 생성되었다.

- 파일 포인터 (LE_7)

데이터 파일을 읽어와서 그래프를 생성하는 과정에서 파일 포인터의 동작 원리를 이해하지 못하여 의도하지 않은 실행 결과가 발생하였다.

- + operator (SE_1)

+ 연산자는 왼쪽과 오른쪽 operand가 문자열인 경우는 양쪽 문자열을 결합하는 용도로, 숫자인 경우는 덧셈으로 사용할 수 있다. 그런데 스크래치 또는 엔트리의 '결합하기' 연산과 혼동하여 문자열과 숫자를 결합하려는 시도로 인하여 오류가 발생하였다.

- 문자열 형식의 숫자 (SE_2)

문자열 형식의 숫자를 숫자로 잘못 판단하여 사칙연산을 수행하고자 하여 오류가 발생하였다.

```
age = '3'
myage = 10 + age
print(myage)
```

Fig. 3. Example of SE_2, not being able to distinguish between a string that looks like a number and a number

예를 들어, Fig. 3에서 age 변수는 '3'이라는 값을 저장한 문자열 형식의 변수인데 10과 더하고자 하는 시도로 인하여 오류가 발생하였다.

- == 와 = 의 구별 (SE_3, SE_4)

변수에 값을 저장하기 위해 = 연산자가 아닌 == 연산자를 사용하거나 두 operand가 같은지 비교하기 위해 == 연산자가 아닌 = 연산자를 사용함으로 인하여 오류가 발생하였다.

- 내장함수와 같은 변수명 (SE_5)

파이썬의 내장함수와 동일한 이름의 변수를 선언하고 사용하여 '~ not callable'을 포함한 오류가 발생하였다.

- 기타 문법 오류 (SE_6)

변수에 아무 값도 저장하지 않은 형태로 프로그램에 사용하는 등 기타 문법적으로 부합하지 않아 발생하는 오류이다.

- CSV 파일의 생성 (DE_1)

엑셀 형식의 데이터를 CSV 형식으로 변환하는 과정에서 발생하는 오류로써 파일의 확장자가 CSV가 아닌 TXT 형식으로 파일을 생성하거나 텍스트 인코딩에 관한 이해 부족으로 오류가 발생하였다.

- 데이터 정제 (DE_2)

데이터 정제 과정에서 발생하는 오류로써 데이터 내에 tab으로 인한 공백과 같은 불필요한 정보들이 존재할 수 있는데 이러한 부분을 완전히 정제하지 않은 상태로 데이터를 읽어와서 사용하려는 과정에서 오류가 발생하였다.

- 파일 오류 (DE_3)

데이터를 파이썬에서 읽어오는 과정에서 파일명을 프로그램내에 명시하였는데 파일명에 오타가 존재하거나 해당 파일 자체가 존재하지 않으므로 파일을 찾지 못하여 오류가 발생하였다.

- 코딩오류 (CE_1, CE_2, CE_3, CE_4)

코딩 과정에서 단순 오타로 인하여 발생하는 오류로써 조건문이나 반복문에 : 을 누락하고 ; 를 사용함에 따른 오류(CE_1), 함수명이나 변수명의 오타(CE_2), 문자열에 ' 또는 " 를 누락함에 따른 오류(CE_3), 그리고 괄호를 누락함에 따른 오류(CE_4) 등이 있었다.

3. Analysis of Error Rate

학생들의 기말 평가 결과물인 파이썬 코드에서 발견한 오류들을 Table 3에서 제시한 오류 타입별로 분류하고 각 항목에 따른 발생 횟수를 Table 4와 같이 정리하였다.

Table 4. Frequency of Errors by Type and Code

Type	Code	Frequency(%)	
Logical Errors	LE_1	3(12.5%)	16(67%)
	LE_3	7(29.2%)	
	LE_4	1(4.2%)	
	LE_5	3(12.5%)	
	LE_6	2(8.3%)	
Syntax Errors	SE_4	1(4.2%)	7(29%)
	SE_6	6(25.0%)	
Coding Errors	CE_1	1(4.2%)	1(4%)

단, 기말 평가에서는 학생들이 초보임을 고려하여 이미 정제가 완료된 CSV 파일을 배포하였으므로 데이터 오류는 분류에서 제외하였다.

오류 타입에 따른 발생 횟수에 의하면, 논리 오류가 67%로 가장 많이 발생하였고 다음으로 문법 오류로 29%였으며 코딩오류는 4%로 상대적으로 적게 발생하였다. 세부 항목별 오류의 발생 횟수에 의하면, 완성된 프로그램의 출력이 원래의 요구 사항에서 일부 기능이 누락되고 부분적인 기능만 구현된 경우(LE_3)가 29.2%로 가장 많이 발생하였다. 두 번째로 많이 발생한 항목은 변수에 아무 값도 넣지 않고 사용하는 등 기타 문법적으로 부합하지 않아 발생하는 오류(SE_6)로써 25%가 발생하였다.

4. Analysis of the Impact of Pre-Learning

초등교사들의 블록형 또는 텍스트형 프로그래밍 언어를 사용한 선행학습이 파이썬 기반의 데이터 분석 프로그래밍 학습에 긍정적인 영향이 있는지 분석하였다. 사전 설문 결과, 전체 21명의 교사 중에서 블록형 프로그래밍 언어를 학습해 본 경험이 있는 경우가 11명(52%), 경험이 없는 경우가 10명(48%)이었다.

먼저, 두 그룹의 정규성 검정(Shapiro-Wilk)을 실시하였으며 Table 5와 같이 정규성이 보장되지 않는 것으로 나타났다($p < 0.05$). 따라서, 블록형 언어의 사전 경험이 파이썬의 학습 효과에 유의미한 차이가 있는지 알아보기 위해 비모수 검정 방법인 Mann-Whitney U Test를 실시하였다.

Table 5. Results of Normality Test(Shapiro-Wilk), block-based programming

Block Programming Experience	<i>n</i>	mean	<i>p</i> -value
Yes	11(52%)	87.6	$p < 0.001$
No	10(48%)	90.3	$p = 0.001$

Table 6. Results of Mann-Whitney U Test, block-based programming

Block Programming Experience	<i>U</i> -value	<i>Z</i> -score	<i>p</i> -value
	53	-0.10	$p = 0.912$

분석 결과는 Table 6과 같이 초등교사들의 블록형 프로그래밍 언어의 선행학습은 파이썬의 학습에 유의미한 영향을 주지 않은 것으로 나타났다($p > 0.05$).

Table 7. Results of Normality Test(Shapiro-Wilk), text-based programming

Text Programming Experience	<i>n</i>	mean	<i>p</i> -value
Yes	6(30%)	100	$p < 0.001$
No	15(70%)	84.4	$p = 0.003$

Table 8. Results of Mann-Whitney U Test, text-based programming

Text Programming Experience	<i>U</i> -value	<i>Z</i> -score	<i>p</i> -value
	15	-2.29	$p = 0.021^*$

* $p < 0.05$

전체 21명의 교사 중에서 텍스트형 프로그래밍 언어를 학습해 본 경험이 있는 경우는 6명(30%), 경험이 없는 경우가 15명(70%)이었다. 동일한 방법으로 정규성 검정(Shapiro-Wilk)을 실시하였으며 Table 7과 같이 정규성이 보장되지 않는 것으로 나타났다($p < 0.05$). 따라서, 텍스트형 언어의 사전 경험이 파이썬의 학습 효과에 유의미한 차이가 있는지 알아보기 위해 마찬가지로 Mann-Whitney U Test를 실시하였다. 분석 결과는 Table 8과 같이 초등교사들의 텍스트형 프로그래밍 언어의 선행학습은 블록형

언어와는 달리 파이썬의 학습에 유의미한 영향을 주는 것으로 나타났다($p < 0.05$).

V. Conclusions

2022년 개정 교육 과정에 따라 최근에는 블록형 뿐만 아니라 텍스트형 프로그래밍 언어의 습득도 요구되기 시작하였으며 이에 따라 현직 초등교사들에게 기대되는 프로그래밍 역량 수준도 높아지고 있다. 본 연구에서는 초등교사들의 프로그래밍 역량 향상을 위해 교사들이 발생시키는 다양한 종류의 오류 데이터를 수집하고 분석하였으며 다음과 같은 결론을 도출하였다.

첫째, 초등교사들이 가장 많이 발생시킨 오류의 타입은 논리 오류였다. 논리 오류는 주어진 과제의 의미를 체계적으로 정리하여 논리적으로 접근하여야 하므로 단순한 문법의 습득보다는 높은 난이도의 프로그래밍 역량이 요구된다. 이러한 이유로 프로그래머가 발생시키는 오류 중에서도 논리 오류가 가장 많은 것으로 보고되었다[9,11,15]. 논리 오류를 감소시키기 위한 방법의 하나는 주어진 과제를 체계적으로 해석하여 단계별로 나누고, 요구 사항을 명확히 정의하고, 이를 알고리즘으로 변환하는 과정을 반복적으로 학습시키면서 체득을 유도하는 것이다. 이때, 스스로 사고하여 해결하는 접근이 익숙하지 않은 학생들은 이 과정에서 흥미를 잃을 수도 있으므로 낮은 수준부터 시작하여 학생들의 수준을 고려하여 점차 난이도를 높여가는 것이 필요하다.

둘째, 초등교사들의 텍스트형 프로그래밍 언어의 선행 학습은 블록형 언어와는 달리 파이썬 기반의 데이터 분석 프로그래밍 학습에 유의미한 영향을 주는 것으로 나타났다. 블록형 언어가 프로그래밍의 기초를 배우기에는 유용하지만 파이썬과 같은 고급 프로그래밍 코드의 작성 시 필요한 깊이 있는 사고를 신장시키기에는 한계가 있기 때문이라고 추측한다. 반면, 텍스트형 언어의 경우는 변수, 데이터 타입, 함수와 같은 다양한 프로그래밍 개념을 미리 경험해볼 수 있고 이를 파이썬에도 그대로 적용할 수 있으므로 이와 같은 결과가 나온 것으로 판단한다. 이러한 결과에 기반하여 향후 파이썬의 교육 시 텍스트형 프로그래밍 언어를 학습한 경험의 여부를 사전에 조사하여 교육의 난이도 조절에 활용할 필요가 있다.

본 연구는 단일 대학에 재학한 21명의 초등교사들을 대상으로 연구를 수행하였기에 결과를 일반화하기에는 무리가 있다. 이에, 향후 다양한 환경의 초등교사들을 대상으

로 오류를 수집하고 분석하여 결과를 도출하고 본 연구의 결과와 비교, 분석해 볼 필요가 있다. 또한, 논리 오류를 줄이는 방안을 교육 현장에 실제로 적용하여 효과에 유의미한 차이가 있는지 분석할 필요가 있다.

REFERENCES

- [1] Ministry of Education, 2022 Revised Curriculum, <https://www.moe.go.kr/boardCnts/viewRenew.do?boardID=141&lev=0&statusYN=W&s=moe&m=0404&opType=N&boardSeq=93458>
- [2] Y.J. Jo and S.J. Chun, "Programming education using Scratch electronic blocks", *Journal of the Korean Association of information Education*, Vol. 24, No. 6, pp. 529-537, Jan. 2020. DOI: 10.14352/jkaie.2020.24.6.529
- [3] K. J. Han, "A Case Study of Elementary School Informatics Education for Pre-service Teachers : Focusing on Programming Education", *Journal of the Korean Association of information Education*, Vol. 25, No. 1, pp. 41-48, Jan. 2021. DOI: 10.14352/jkaie.2021.25.1.41
- [4] K. Hur, "A Study on Elementary Education Examples for Data Science using Entry", *Journal of the Korean Association of information Education*, Vol. 24, No. 5, pp. 473-481, Oct. 2020. DOI: 10.14352/jkaie.2020.24.5.473
- [5] J.S. Yu and M.H. Lee, "Effects of a Programming Class Using Dolittle on Enhancing Creativity, Problem Solving Ability, and Interest in Programming", *Journal of the Korean Association of information Education*, Vol. 13, No. 4, pp. 443-450, Dec. 2009. UCI: G704-000854.2009.13.4.001
- [6] K.J. Park and S.J. Lee, "A Comparative Study of the Effect of Dolittle and Robot Programming Education on Creativity", *Journal of the Korean Association of information Education*, Vol. 14, No. 4, pp. 619-626, Dec. 2010. UCI: G704-000854.2010.14.4.014
- [7] J.Y. Lee, "Effects of Data Visualization Education Using Python on Improvement of Computational Thinking Ability in Information Gifted Students of Elementary school", Master's thesis, Korea National University of Education. Chung-Buk, Korea, 2021.
- [8] M.H. So and J.M. Kim, "Transference from learning block type programming to learning text type programming", *The Korean Association of Computer Education*, Vol. 19, No. 6, pp. 55-68, Nov. 2016. DOI: 10.32431/kace.2016.19.6.006
- [9] W.S. Moon, "Analysis of error data generated by prospective teachers in programming learning", *Journal of the Korean Association of information Education*, Vol. 22, No. 2, pp. 205-212, April 2018. DOI: 10.14352/jkaie.2018.22.2.205
- [10] J.W. Nam and I.H. Yoo, "Development of NXC Robot Programming Supporting System Based on Types of Programming Error", *Journal of the Korean Association of information Education*, Vol. 15, No. 3, pp. 375-385, Sep. 2011. UCI: G704-000854.2011.15.3.002
- [11] J.S. Kim and Y.S. Kim, "The Analysis of Relationship between Academic Achievement Level of Concept Learning and Error Type in Online Programming Course", *The Korean Association of Computer Education*, Vol. 17, No. 5, pp. 43-51, Sep. 2014. DOI: 10.32431/kace.2014.17.5.005
- [12] H.W. Jung, "A Case Study of Python Programming Error in an Online Learning Environment". *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 7, No. 3, pp. 247-253, Aug. 2021. DOI: 10.17703/JCCT.2021.7.3.247
- [13] A.J. Ko and B.A. Myers, "A framework and methodology for studying the causes of software errors in programming systems", *Journal of Visual Languages & Computing*, Vol. 16, Issues 1-2, pp. 41-84, February-April 2005, DOI: 10.1016/j.jvlc.2004.08.003
- [14] A.J. Ko and B.A. Myers, "Development and evaluation of a model of programming errors", In *IEEE Symposium on Human Centric Computing Languages and Environments*, Oct. 2003, Proceedings, pp. 7-14, IEEE.
- [15] A. Ettles, A. Luxton-Reilly, and P. Denny, "Common logic errors made by novice programmers", In *Proceedings of the 20th Australasian Computing Education Conference*, pp. 83-89. Jan. 2018.

Authors



Junghee Jo received the Ph.D. degree in Computer Science from University of Massachusetts Amherst, USA, in 2014. She had worked at LG Electronics and Electronics and Telecommunications Research

Institute (ETRI), respectively. She is currently an associate professor in the Department of Computer Education, Busan National University of Education, Korea. Her research interests include computer education, human factors, and health informatics.