

수행시간 중심의 초등 EPL 교육이 초등학생 컴퓨팅 사고력에 미치는 효과

문우종* · 부용호** · 김종훈***

보목초등학교*, 송당초등학교**, 제주대학교***

요약

본 연구는 초등학생의 컴퓨팅 사고력 신장을 위한 교육 방법으로 수행시간 중심의 초등 EPL 교육 프로그램을 개발하여 적용한 후 그 효과를 검증하였다. 교육 프로그램은 초등학교 학생 48명을 대상으로 실시한 사전 요구 분석 결과를 바탕으로 하여 개발하였다. 개발한 교육 프로그램의 효과를 검증하기 위해 ○○대학교에서 실시한 초등학생 교육 기부 프로그램의 지원자 표집에 의한 지원자 표본 25명을 대상으로 6일간 1일 7차시씩 42차시를 진행하였다. 한국정보과학교육연합회에서 주최한 '비버챌린지(Korea Bebras Challenge 2018)'를 활용하여 사전·사후 검사 결과를 통해 교육적 효과를 분석하였으며, 분석 결과 수행시간 중심의 초등 EPL 교육이 초등학생의 컴퓨팅 사고력 향상에 긍정적인 영향을 미친 것으로 나타났다.

키워드 : 알고리즘 교육, 알고리즘 수행시간, EPL 교육, 스크래치, 컴퓨팅 사고력

Effect of Execution Time-oriented the Elementary EPL Education on Computational Thinking Ability of Elementary School Students

Woojong Moon*, YongHo Boo**, Jonghoon Kim***

Bomok Elementary School*, Songdang Elementary School**, Jeju National University***

ABSTRACT

This study has developed and applied an elementary EPL education program centered on execution time as an educational method for improving Computational thinking of elementary school students, and then its effect has been verified. The education program was developed based on the results of preliminary demand analysis conducted on 48 elementary school students. In order to verify the effectiveness of the education program, 25 students were asked to participate in the class of elementary school students on a seven hours for six days. Using the "Korea Bebras Challenge 2018", educational effects were analyzed through the results of pre- and post-mortem examinations, and the analysis showed that elementary EPL education centered on performance time was effective in improving computational thinking of elementary school students.

Keywords : Algorithm, Algorithm Execution time, EPL education, Scratch, Computational thinking

본 논문은 2020년도 제주대학교 교원성과지원사업에 의하여 연구되었음.

교신저자 : 김종훈(제주대학교 초등컴퓨터교육전공)

논문투고 : 2020-05-21

논문심사 : 2020-06-01

심사완료 : 2020-06-04

1. 서론

4차 산업혁명의 도래로 미래사회에 대비한 교육이 어떻게 이루어져야 하는지에 대한 논의가 어디서나 활발하게 이루어지고 있다. 2017년 국제미래학회에서 발간한 제4차 산업혁명 시대 대한민국 미래 보고서에 따르면 교육의 방향은 첨단 기술의 발전에 대비하여 다양성, 창의성, 유연성을 기르는 방향으로 변화시켜 나가야 한다[2].

우리나라에서도 이러한 시대적인 흐름에 따라 2015 개정 교육과정에서 소프트웨어 교육을 본격적으로 도입하였고, 초·중등학교에서 언플러그드 활동, 프로그래밍 교육, 피지컬 컴퓨팅 활용 교육 등 다양한 형태로 소프트웨어 교육이 이루어지고 있다. 초등학교에서는 대부분 블록형 프로그래밍 언어를 활용한 기초 프로그래밍 교육을 실시하고 있는데, 이는 초등학생들의 인지적 수준에 적합하고 초등학생들의 흥미와 관심을 끌 수 있는 시청각적인 요소를 제시하고 있기 때문이다[4].

그러나 학생들이 프로그래밍 학습 과정에서 개발하고 있는 컴퓨팅 사고력에 대한 평가는 아직 활발하게 이루어지지 못하고 있다. 그 이유로는 컴퓨팅 사고력 평가의 중요성에 대한 인식의 부재, 교수자와 학습자가 손쉽게 접근할 수 있는 컴퓨팅 사고력의 평가 도구의 부재, 컴퓨팅 사고력의 평가 방법에 대한 복잡성 등이 있다[8]. 소프트웨어 교육 운영지침(2015)에서도 소프트웨어 교육의 목표를 '컴퓨팅 사고력을 가진 창의·융합 인재 양성'으로 기술하고 있는 만큼, 컴퓨팅 사고력을 효과적으로 향상시키기 위한 연구가 필요하다[9].

이에 본 연구에서는 초등학생의 컴퓨팅 사고력 향상을 위해 알고리즘 수행시간을 주제로 한 초등 EPL 교육 프로그램을 개발하여 적용하였다. 개발한 교육 프로그램은 사전 요구 분석을 통해 교육 내용을 선정하였으며 초등학교 4~5학년 학생들 중 지원자표본 25명의 학생을 대상으로 투입하였다. 교육 프로그램 실시 후 컴퓨팅 사고력 검사 도구로 비버챌린지(Korea Bebras Challenge 2018)를 사용하여 향상 정도를 평가하였으며, 컴퓨팅 사고력 향상을 위한 효과적인 EPL 교육 방법을 제시하고자 한다.

2. 이론적 배경

2.1 EPL 교육

EPL(Educational Programming Language)은 교육용 프로그래밍 언어로 학생들이 프로그래밍을 접하는 데 있어 부담을 최소화 할 수 있도록 고안된 컴퓨터 언어이다. 일반적인 프로그래밍 언어는 기본 문법이나 구조를 익히고 구문 오류를 해소하는데 많은 시간과 인지적 부담을 초래한다. 교육용 프로그래밍 언어는 어린 학습자나 초보 학습자가 다루기 쉬운 형태로 구성되어 있어 프로그래밍 학습 과정에서 부가되는 학습자들의 인지적 부담을 감소시켜 주고, 프로그래밍 학습에 흥미를 유발하여 몰입하게 한다[3][11]. 본 연구에서는 알고리즘 수행시간을 중심으로 초등학생의 컴퓨팅 사고력 향상을 위하여 EPL 교육 방법을 채택하였으며, EPL 중에서 가장 보편적이고 많은 장점을 지닌 스크래치를 사용하였다.

2.2. 스크래치

스크래치 프로그램은 2007년 미국의 MIT Media Lab에서 8세에서 16세 정도의 어린이의 지능과 창의성 능력 개발을 위해 만들어진 무료 프로그래밍 도구이다. 생각을 실천하고 공유하는 과정을 지속적으로 반복하도록 하여 어린 학생들이 창의성 습관을 키울 수 있도록 만들어졌으며, 인터프리터 방식의 교육용 프로그래밍 언어이다. 스크래치는 아이들이 생각하는 알고리즘을 블록이라는 내장된 논리적 프로그래밍을 서로 결합하는 방식을 채택하여 기존의 텍스트 입력 방식이 가지고 있는 코딩과 문법의 어려움 등을 극복하였으며, 다양한 멀티미디어(그래픽, 소리 등)를 혼합하여 프로그래밍을 할 수 있도록 설계되어 있다. 게다가 용량이 매우 작아 소프트웨어 휴대가 편하며 만든 프로그램을 웹사이트를 통해 손쉽게 공유하고 오프라인 에디터를 통해 인터넷 없이도 실행이 가능하다. 이에 본 연구에서는 가장 최신 버전인 스크래치 오프라인 에디터 3.0 버전을 선정하여 본 교육에 활용하였다[15][16].

2.3 알고리즘 수행시간

알고리즘 수행시간은 입력(input)에 따라 알고리즘이 실행되어 출력(output)되기까지 소요되는 시간을 의미한다. 알고리즘의 성능은 수행시간과 필요한 기억 장치의 양에 따라 평가할 수 있는데 대부분 수행 시간이 더 중요한 의미를 지닌다. 알고리즘 수행시간을 분석하여 계산한 결과를 시간 복잡도(Time Complexity)라고 하며, 알고리즘의 핵심이 되는 연산의 횟수를 세어 성능을 파악할 수 있다. 알고리즘 수행시간에 따라 알고리즘 내의 연산 횟수가 크게 달라지기 때문에, 효율적인 문제 해결을 위해서는 알고리즘 수행시간에 초점을 두어야 한다[7]. 본 연구에서는 수행시간을 학생들이 쉽게 비교하기 위해 스크래치 프로그램 내에서 계산을 하는 부분이 몇 번 반복되는지 그 횟수를 세는 것으로 초등학생 수준에 맞춰 단순화하여 지도하였다. 학생들은 알고리즘 수행시간을 최소화하는 알고리즘을 찾기 위해 다양한 시도를 해보는 과정에 초점을 두었다.

2.4 컴퓨팅 사고력

컴퓨팅 사고력은 Wing(2006)이 ‘해결해야 할 문제를 만났을 때 컴퓨터 과학자처럼 사고하는 것’이라고 정의하며 ‘Computational Thinking’이라는 용어를 소개하는 과정에서 등장하였다[13]. 용어가 국내에 도입되면서 초기에는 연구자에 따라 ‘계산적 사고’, ‘컴퓨터 과학적 사고’, ‘컴퓨팅적 사고’ 등으로 다르게 사용되다가, 한국과학창의재단(2014)에서 혼용되고 있는 용어를 통일하고 교육 현장에 효과적으로 도입하고자 ‘Computational Thinking’을 ‘컴퓨팅 사고력’이라는 용어로 통일시켰다. 그 결과 교육부의 소프트웨어 교육 운영지침(2015)에서도 소프트웨어 교육의 목표를 ‘컴퓨팅 사고력을 가진 창의·융합 인재 양성’으로 기술하며 컴퓨팅 사고력이 보편화 되었다[9].

컴퓨팅 사고력의 정의는 학자마다 다르나 문제를 수립하고 해결책을 찾아 컴퓨팅 시스템을 통해 효과적으로 수행되도록 표현하는 사고 과정으로서, 추상화와 자동화를 요소로 문제를 해결하는 절차적 사고 능력으로 종합해볼 수 있다. 컴퓨팅 사고력은 21세기의 모든 학습자들이 갖추어야 할 3R(읽기, 쓰기, 셈하기)에 더하여

기본적으로 갖추어야 할 능력으로 평가받고 있으며, 소프트웨어 교육의 목표는 컴퓨팅 사고력을 기르는 것이라고 정리할 수 있다[12]. 본 연구에서는 문제를 해결하는 수행시간을 EPL을 통해 비교·분석하고 더 효율적인 알고리즘을 찾아내는 과정에서 컴퓨팅 사고력의 신장을 목표로 하였다.

2.5 선행연구 분석

윤선희(2018)의 연구에서는 EPL이 프로그래밍 교육의 프로그래밍, 컴퓨팅 사고력 및 문제해결력에 미치는 영향에 대해 분석하였다. EPL 중 스크래치를 병행하여 프로그램 언어 교육을 지도한 결과 학생들의 프로그래밍 및 컴퓨팅 사고력과 문제해결력 향상에 도움이 되었으며, 학생들의 만족도 또한 높은 것으로 나타났다[14].

임서은(2017)의 연구에서는 컴퓨팅 사고력 향상을 위한 알고리즘 기반의 교수·학습 방법을 개발하였다. 정보교과의 컴퓨팅 과학 문제를 해결하기 위해 알고리즘 기반의 교수학습 방법을 활용함으로써 학생들이 문제를 해결하는 과정에서 컴퓨팅 사고력이 향상되었다[10].

박주연(2019)의 연구에서는 초등학생의 스크래치 프로젝트의 코드를 분석하는 방법을 활용하여 컴퓨팅 사고력의 개념 수준을 평가하였다. 연구 결과 초등학생들은 컴퓨팅 사고력의 요소 중 소프트웨어 교육의 핵심인 논리와 추상화가 가장 낮은 것으로 나타났다. 초등학생의 수준에 논리와 추상화는 어려운 개념일 수 있으나 학생들의 인지 수준에 맞도록 구체적이고 단계적으로 접근하여 교수가 필요함을 제안하였다[8].

양영훈(2019)의 연구에서는 초등학생을 대상으로 정렬, 탐색 알고리즘 수행시간 분석을 주제로 한 교육 프로그램을 개발하여 적용하고 교육적 효과를 분석하였다. 연구 결과 알고리즘 수행시간을 비교하고 분석하는 교육을 통해 초등학생의 논리적 사고력 향상에 긍정적인 영향을 주었으며, 수행시간이 사고력 향상을 위한 효과적인 프로그래밍 교육 주제가 될 수 있음을 제안하였다[17][18].

위에서 살펴본 것처럼 소프트웨어 교육의 본질적인 목표인 컴퓨팅 사고력의 신장을 위해 스크래치, 파이썬 등의 프로그래밍 언어를 활용한 교육 연구가 최근까지도 많이 이루어지고 있다. 본 연구에서는 학생들의 컴퓨

팅 사고력 향상을 목표로 알고리즘 수행시간을 주제로 선정하고, 학생들의 인지적 수준을 고려하여 수행시간의 개념을 파악하기 쉽도록 단순화하여 제시하였다. 또한 주제의 난이도를 고려하여 요구 분석 결과 학생들에게 친숙한 EPL중 스크래치를 교육 도구로 선정하고, 개발한 교육 프로그램이 초등학생의 컴퓨팅 사고력 향상에 미치는 영향을 검증하고자 하였다.

3. 연구 방법

3.1 연구 가설

귀무가설: 수행시간 중심의 초등 EPL 교육에 의해 학습자의 사전, 사후 컴퓨팅 사고력은 차이가 없다.

대립가설: 수행시간 중심의 초등 EPL 교육에 의해 학습자의 사전, 사후 컴퓨팅 사고력은 차이가 있다.

3.2 연구 대상

본 연구의 대상은 ○○대학교에서 실시한 교육 기부 프로그램의 지원자 표집에 의한 지원자 표본(volunteer sample) 25명을 선정하였다. 연구 대상의 학년과 성별은 <Table 1>과 같다.

<Table 1> Grade and gender of the subject

class	Male	Female	Total
4 Grade	11	6	17
5 Grade	4	4	8
Total	15	10	25

3.3 교육 프로그램

본 연구에서는 교수 체제 설계 과정의 일반적 형태를 나타내어 가장 널리 활용되고 있는 Dick& Carey의 ADDIE 모형의 절차에 따라 교육 프로그램을 <Table 2>와 같이 개발하였다[5].

<Table 2> Educational Program Development Plan

Analysis	· Demand Analysis (for elementary school students)
Design	· Specification of performance objectives - Effect of Execution Time-Oriented Elementary EPL education · Teaching Strategy
Development	· Development of Teaching Materials - 42nd-hour plan - Teaching Materials
Implementation	· SW Training using EPL (focusing on execution time-oriented using Scratch)
Evaluation	· Post-Evaluation with Korea Bebras Challenge and Performance check

3.3.1 요구 분석

ADDIE 모형의 절차에 따른 요구 분석은 제주도 내 초등학교 고학년 학생 48명을 대상으로 한 설문 조사 결과를 바탕으로 실시하였다.

<Table 3> The experience of SW education

	Unplugged	Physical Computing	EPL	Computer Language
student	11(23%)	20(42%)	16(33%)	1(2%)

학생들이 소프트웨어 교육에서 주로 경험한 수업 방법 결과는 <Table 3>과 같다. 조사 결과 피지컬 컴퓨팅이 가장 많았으며 EPL, 언플러그드, 컴퓨터 언어 순으로 교육이 이루어지고 있었다.

<Table 4> Understanding Algorithms in SW education

	not well aware	be average	well-informed
student	9(18.8%)	21(45.8%)	17(35.4%)

소프트웨어 교육에서 알고리즘(절차적 문제 해결)을 알고 있는지 조사한 결과는 <Table 4>와 같았다. 80% 이상의 학생들은 소프트웨어 교육을 통해 알고리즘이라는 용어를 접해보았으며, 알고리즘의 의미를 이해하고 있다고 응답하였다.

<Table 5> The need of Algorithms in SW education

	Do not need	be Normal	Essential
student	4(4%)	19(19%)	77(77%)

소프트웨어 교육에서 알고리즘 교육이 필요한가라는 질문에 96%의 학생은 <Table 5>와 같이 알고리즘 교육이 꼭 필요하다고 응답하였다.

요구 분석 결과를 분석한 결과 다음과 같은 요구를 도출할 수 있다.

첫째, 학생들이 주로 경험하는 소프트웨어 교육 방법은 피지컬 컴퓨팅, EPL, 언플러그드 순이며 컴퓨터 언어는 거의 경험하지 못하였다. 알고리즘 수행시간이라는 주제가 다소 생소한 만큼 학생들이 자주 접해본 도구 중에서 수행시간을 비교 분석하기 편리한 EPL(스크래치)을 본 수업 도구로 선정하였다.

둘째, 대다수의 학생들은 소프트웨어 교육을 통해 알고리즘(절차적 문제 해결)을 접해보았으며 어느 정도 이해하고 있다고 답변하였다. 더불어 알고리즘 수행시간(문제를 해결하는 방법이 효율적인지 알아보는 방법)에 대한 교육이 필요하다고 공감하였다. 이에 학생들이 동일한 문제를 해결하는 여러 가지 알고리즘을 수행시간 면에서 스크래치를 활용해 분석해보고, 어떤 알고리즘이 더 효율적인지 파악하는 과정에서 학생들의 컴퓨팅 사고력 향상을 꾀하는 데 중점을 두었다.

3.3.2 프로그램 설계 및 개발

학습자들의 수준과 스크래치의 특성을 고려하여 수행시간 중심의 초등 EPL 교육 학습 내용을 구성하였다. 초기에는 스크래치의 기초 UI와 반복문, 조건문, 변수 등 알고리즘 수행시간을 비교하고 분석하는 데 필요한 내용을 중점으로 설계하였으며, 해당 내용을 익숙하게 다룰도록 개별 과제를 제시하였다. 중반부터는 만보기, 열량 계산기, 소수 판별하기, 에라토스테네스의 체 등 여러 가지 알고리즘으로 문제를 해결해보고 각각의 수행시간을 비교해보며 더 효율적인 알고리즘을 찾도록 구성하였다. 후반에는 스스로 주제를 선정하여 두 가지 이상의 알고리즘으로 문제를 해결하고 각각의 수행시간을 비교하는 개별 프로젝트를 진행하였다. 이에 따른 차시별 학습 내용은 <Table 6>과 같다.

<Table 6> The Theme of education program

Hour	Learning theme
1~8	- Introduction to Scratch, Installation and UI Guide - Scratch Foundation
9~16	- Repetitive Statement, Conditional Statement - Variable, List, Eratosthenes' sieve
17 ~ 24	- Comparison of algorithm execution time(1) (through pace counter, common measure)
25 ~ 32	- Comparison of algorithm execution time(2) (through Euclidean algorithm, prime number)
33 ~ 42	- Individual Project Challenges - Present individual project work

(Fig.1)은 최대공약수 구하기를 주제로 수행시간의 개념(알고리즘 혹은 프로그램이 입력을 받은 후 출력을 내고 종료하기까지 걸린 시간)을 도입하고 알고리즘 내에서 수행시간을 어떻게 측정(프로그램 내에서 계산을 몇 번 하는지 횟수를 세기)하는지 설명하는 교재의 일부이다[6].

Execution Time-oriented the Elementary EPL education

Chapter 9. Finding the Greatest Common Denominator

1. The Greatest Common Denominator: the largest of two or more common divisor

2. Find The Greatest Common Denominator

Variable Settings: There are 'big' and 'small' of the two numbers to obtain the maximum common divisor. And set the variable up to the 'number' that should be divided. It also creates a 'Greatest Common Denominator'.

The 'number' you divide begins with 1 and repeats it by increasing it to 'number' by using 'switch 1'.

Repeat until 'su' becomes 'minor' and store the value of 'su' in 'Maximum Promise' if the remainder of 'minor'÷'su' is zero. This clears the number previously stored in 'Maximum Committee' and stores the new value. Finally, say the data stored in the 'Maximum Committee'.

3. What is Execution time-oriented?
Execution time-oriented is the time it takes for an algorithm or program to receive input, output, and shutdown, and the shorter the performance, the shorter the processing time and the more can be solved. You can count the number of times you do the calculation in the program.

4. What determines the Execution time-oriented? Let's draw a graph.

ScrATCH

(Fig. 1) Example of educational program

(Fig.2)는 같은 주제를 두 개의 프로그램으로 해결한 알고리즘의 수행시간(프로그램 내 계산 횟수)을 비교해보는 교재의 일부이다[6].

Execution Time-oriented the Elementary EPL education

Chapter 11. To distinguish Prime Number

1. What is prime number?
Prime Number is a natural number larger than 1 and cannot be divided into natural water other than itself. In other words, it's only one mineral water and one self-confident number. As in 2, 3, 5, and 7, the numbers with only one mineral water are called the few.

2. To distinguish Prime Number

<Method 1>
If there are two falls from 1 to yourself, it is a prime number, otherwise it is not a prime number.

<Method 2>
From 2 to 1, it is a prime number if it does not fall, and if it falls, it is not a prime number.

3. Let's compare the performance time of method 1 and method 2.

method 1	method 2

ScrATCH

(Fig. 2) Example of educational program

3.4 연구 도구

연구 도구로는 컴퓨팅 사고력의 향상을 점검할 수 있도록 한국정보과학교육연합회에서 주최한 비버챌린지 (Korea Bebras Challenge 2018) 검사지를 사용하였다. 비버챌린지는 정보과학의 개념을 기반으로 알고리즘과 프로그래밍, 자료 분석과 자료 표현, 컴퓨팅 시스템의 구성과 동작 원리를 주제로 문제 분석, 핵심요소 추출, 문제 분해, 모델링 등 컴퓨팅 사고력 요소들을 평가하고 측정할 수 있는 검사 도구이다[1]. 본 연구에서는 비버챌린지 2018 검사 중 교육에 참여하는 학생들의 연령과 수준을 고려하여 비버챌린지 그룹Ⅲ(초등학교 5~6학년용) 검사지를 선정하고, 교육 사전-사후 검사지로 활용하여 컴퓨팅 사고력의 향상 정도 변화를 측정하였다.

3.5 연구 설계 및 처치

본 연구에서는 수행시간 중심의 초등 EPL 교육을 주제로 초등학교의 컴퓨팅 사고력 향상을 위한 교육 프로그램을 개발하여 적용하였다.

연구 대상은 초등학교 4~5학년 25명의 단일집단으로 선정하였으며, 교육 프로그램은 요구 분석에 따라 내용을 체계화하였고 1일 7차시씩 총 42차시의 강의와 실습으로 구성하였다. 실제 교수·학습은 80분 단위의 블록 수업으로 진행하였으며, 학습의 효과를 극대화하기 위해 6일간 연이어 집중 이수의 형태로 교육을 진행하였다.

교육 프로그램의 효과를 분석하기 위해 컴퓨팅 사고력 검사 도구인 비버챌린지를 사용하여 단일집단 전후 검사를 실시하였다.

이 설계 과정을 도식화하면 <Table 7>과 같다.

<Table 7> Experimental design

	Pre-test	Treatment	Post-test
G	O1	X	O2

G: Experimental group(N = 25)

O1, O2: Bebras Challenge 2018(Figure A, B style)

X: Software education program

4. 연구 결과

4.1 교육 프로그램 효과 검증

수행시간 중심의 초등 EPL 교육이 학생들의 컴퓨팅 사고력에 어떤 영향을 주었는지 분석하기 위한 검증을 실시하였다.

4.1.1 컴퓨팅 사고력 검사 정규성 검증

표본의 크기가 25명으로 '10≤n<30'에 위치하기 때문에, 실험 집단의 컴퓨팅 사고력 검사 결과가 정규성을 확보하고 있는지 정규성 검정을 시행하였다. 정규성 검정은 샤피로-윌크(Shapiro-Wilks) 검사를 실시하여 그 결과를 <Table 8>에 제시하였다.

<Table 8> Normality test

Subscales	Descriptive Statistics(N=25)				stat	p
	M	SD	Max	Min		
CT	1.240	1.200	3	-1	.920	.282

*p<.05

정규성 검증 결과 유의확률(p)이 '0.282'로 0.05보다 크게 나타나 귀무가설을 채택하여 정규성을 만족한다고 가정하였다.

4.1.2 컴퓨팅 사고력 사전·사후 검사 비교

컴퓨팅 사고력 검사 결과가 샤피로-윌크(Shapiro-Wilks) 검사 결과 정규성을 확보하여, 사전·사후 검사 결과 비교는 대응표본 t검정(Paired T-test)을 실시하였다.

<Table 9> Changes in Computational Thinking(Paired T-test)

Subscales	N	Pre-Test		Post-Test		t	p
		M	SD	M	SD		
CT	25	5.28	.363	6.32	.298	-3.436	.002*

*p<.05

대응표본 t검정 결과인 <Table 9>에서 평균점수는 사전 5.28에서 사후 6.32로 1.08점 상승하였고, 유의확률은 0.002로 컴퓨팅 사고력이 교육 전후 통계적으로 유의미하게 향상된 것으로 나타났다.

4.2 연구 결과 분석

비버켈린지 결과를 분석하기 전 실험집단의 검사 결과가 정규분포를 갖추었는지 확인하기 위해 샤피로-윌크(Shapiro-Wilks) 검사를 실시하였다. 검사 결과 정규성을 만족한다고 가정하고 사전-사후 검사 결과 비교를 위해 대응표본 t검정(Paired T-test)을 실시하였다.

대응표본 t검정 결과 컴퓨팅 사고력은 사전 검사 대비하여 평균점수도 상승하였으며, 유의확률도 통계적으로 유의미한 향상을 보였다.

따라서 본 연구를 통해 수행시간 중심의 초등 EPL 교육이 초등학생의 컴퓨팅 사고력을 향상시킬 수 있음을 입증하였으며, 수행시간이라는 주제로 초등학생의 소프트웨어 교육 주제로 충분히 유용하게 활용될 수 있음을 제시하였다.

5. 결론

본 연구에서는 수행시간 중심의 초등 EPL 교육이 초등학생들의 컴퓨팅 사고력에 미치는 영향을 파악하고자 하였다. 이를 위해 ADDIE 모형의 절차에 따라 사전 요구 분석을 토대로 교육 프로그램을 개발하고 6일간의 집중 이수 교육을 실시한 뒤 사전·사후 검사를 통해 컴퓨팅 사고력의 변화를 검증하였다.

검사 결과 본 연구에서 개발한 교육 프로그램은 초등학생의 컴퓨팅 사고력 향상에 효과적인 것으로 나타났으며, 학생들은 수행시간을 비교하는 과정에서 다양한 시도를 해보며 프로그래밍 실력이 많이 향상되는 등 교육의 만족도도 높았다. 이에 알고리즘 수행시간 비교·분석이라는 주제는 소프트웨어 교육의 목표인 컴퓨팅 사고력을 기르는 데 충분한 가치를 지닌다고 볼 수 있다.

다만, 본 연구의 실험집단은 일반적인 상권 연구에 필요한 30명 이상의 참여자를 확보하지 못하여 일반화하는 데 한계가 있다. 더불어 본 연구에서 개발한 프로그램은 비교집단 없이 실험집단의 사전·사후 검사 결과

만으로 검증이 이루어졌기 때문에, 본 프로그램의 영향 때문인지 상관관계를 분명하게 규정할 수 없다는 문제가 있다.

추후의 연구에서는 다수의 참여자를 확보하고, 실험 집단과 비교집단을 구성하여 연구 결과에 대한 각 요인들을 좀 더 체계적으로 분석할 필요가 있다.

참고문헌

- [1] Bebras Challenge(2018). Introduction to Bebras Challenge. <http://bebras.org>
- [2] Daesik Kim(2017). Identity and Role of Elementary Education in the Fourth Industrial Revolution Era. *The Journal of Korean Educational Idea*, 31(4), 23-45.
- [3] Eunkyong LEE(2008). The Effects of a Robot Based Programming Learning on Learners' Creative Problem Solving Potential. *Journal of the Korean Institute of industrial educators*, 33(2), 120-136.
- [4] Jaekwoun Shim, Jeongmin Chae(2018). Development of On-line Judge System based on Block Programming Environment. *The Journal of Korean Association of Computer Education*, 21(4), 1-10.
- [5] Jiyeon Lee(2008). Practical Teaching Methods and Educational Engineering for Prospective Teachers, 1st Ed. Seohyun publishing co.
- [6] Jonghoon Kim(2014). Scratch Algorithm. Hanbit Academy publishing co.
- [7] Jongsun Hwang, Namyong Kim, Dongchul Shin(2005). Computer Internet IT Dictionary, 1st Ed. Iljin publishing co.
- [8] Juyeon Park(2019). Evaluation of Computational Thinking through Code Analysis of Elementary School Students' Scratch Projects. *Journal of The Korean Association of Information Education*, 23(3), 207-217.
- [9] Ministry of Education(2015). Software Education Guideline.
- [10] Seo Eun Lim(2017). Development of Teaching and Learning Methods Based on Algorithms for Improving Computational Thinking. *Journal of The Korean Association of Information Education* 21(6), 629-638.
- [11] Seongkyun Jeon(2014). Art based STEAM Education Program using EPL. *Journal of The Korea Society of Computer and Information* 19(4), 149-158.
- [12] Seungwon Lee(2019). The Effect of Self-regulating Learning-Based Unplugged Activities on Computing Thought in Primary Room and SW Education. Chungnam National University. Doctor's Thesis.
- [13] Sookyoung Choi(2011). An Analysis of "Informatics" Curriculum from the Perspective of 21st Century Skills and Computational Thinking. *Journal of The Korean Association of Computer Education* 14(6), 19-30.
- [14] Sunhee Yoon(2018). A Study on the Effect of EPL on Programing, Computing Thinking and Problem Solving Ability of Programing Education. *The Journal of the Convergence on Culture Technology* 4(4), 287-294.
- [15] Waeshik Moon(2014). Development and Application of STEAM Education Model using Scratch Programming and Sensor Board in Class of Elementary School Students. *Journal of The Korean Association of Information Education*, 18(2), 213-224.
- [16] Wiki(2020). Introduction to Scratch. <http://ko.wikipedia.org/wiki>
- [17] Yeonghoon Yang, Gipyong Kong(2019). Effect of search algorithm execution time analysis education on logical thinking of elementary school students. *Journal of The Korean Association of Information Education*, 23(2), 179-188.
- [18] Yeonghoon Yang, Woojong Moon(2019). Effect of Execution Time-oriented Python Sort Algorithm Training on Logical Thinking Ability of

Elementary School Students. *Journal of The Korean Association of Information Education*, 23(2), 107-116.

저자 소개

문 우 종



2019~ 제주대학교 과학교육학부
컴퓨터교육과 박사과정

2018~ 현재 보목초등학교 교사
관심분야 : SW교육, 프로그래밍

E-Mail : mwj1006@korea.kr

부 용 호

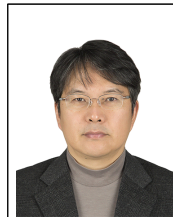


2018~ 제주대학교 교육대학원
초등컴퓨터교육과 수료

2020~ 현재 송당초등학교 교사
관심분야 : SW교육, 메이킹

E-Mail : bu005@naver.com

김 종 훈



1999~ 현재 제주대학교 교육대
학 초등컴퓨터교육전공 교수

관심분야: 컴퓨터교육

E-Mail: jkim0858@jejunu.ac.kr