

초등학교 소프트웨어교육에서 교구와 학습자 수준에 따른 컴퓨팅사고력 향상도 비교

Comparison of Computational Thinking Improvement Based on Teaching Aids and Student's Level in Elementary Software Education

이영재[†] · 김영식^{††}

YoungJae Lee[†] · Yungsik Kim^{††}

요 약

소프트웨어교육의 중요성이 강조되면서, 초등학교의 컴퓨팅사고력 향상을 위한 다양한 교육이 이루어지고 있다. 하지만 선행연구들을 살펴보면 특정 환경이나 학습자만을 고려한 경우가 많아 학교현장에 적용하는 것에 한계가 있다. 따라서 본 연구에서는 교과서의 내용을 바탕으로 학습자에게 영향을 미칠 수 있는 요인인 교구와 컴퓨팅 사고력에 대한 학습자 수준이라는 특성을 고려한 교수·학습활동을 구성하여 소프트웨어교육을 실시하고, 학습자들의 컴퓨팅사고력 향상도 비교를 통해 효과적인 소프트웨어교육 방법의 방향을 제안하였다. 연구결과 교구와 학습자 수준에 따라서 컴퓨팅사고력 향상에 차이가 있는 것으로 나타났는데 구체적으로 활용 교구에 있어서는 비주얼 프로그래밍과 언플러그드 방식을 활용한 교수·학습활동이 보다 효과적이었으며 학습자 수준에 따라서는 사전검사에서 컴퓨팅사고력의 수준이 낮은 것으로 평가된 학생들이 상대적으로 높은 향상도를 보였다. 이러한 경향은 초등학교 소프트웨어교육에서 교수·학습활동을 구성함에 있어 다양한 학습요인을 고려할 필요성이 요구된다고 보인다.

주제어: 초등학교 소프트웨어교육, 컴퓨팅사고력, 교수·학습활동, 교구, 학습자 수준

ABSTRACT

As the importance of software education is emphasized, various educations for improve the computational thinking of elementary school students are being provided. However, there are many limitations that apply to the school at preceding studies, because they were invented for specific environment or students. Therefore, in this study, software education was conducted by constructing teaching-learning activities in consideration of the characteristics of students such as teaching aids and level of computational thinking ability, which can influence students based on the contents of textbooks. Through the comparison about computational thinking improvements, this study tried to suggest the direction of effective software education method. As a result, there are differences in computational thinking improvements according to the student's level and used teaching aids. Specifically, visual programming and unplugged type teaching-learning activities were more effective in teaching aids, and according to the level of students, students with low level of computing thinking in pre-inspection showed relatively high improvement. This tendency suggests that it is necessary to consider various learning factors in designing teaching-learning activities in elementary software education.

Keywords: Elementary software education, Computational thinking, Teaching-learning activity, Teaching aids, Student's level

1. 연구의 필요성

4차 산업혁명과 정보화의 약진이라는 세계적인 변화를 반영하여 컴퓨팅사고력과 정보문화소양, 협력적 문제 해결력을 키우기 위한 교육이 전 세계적으로 실시되고

있다. 우리나라 또한 2015 개정 교육과정에 따라 초·중·고 전 학교에 관련 교육이 적용되고 있으며, 특히 초등학교 단계에서는 건전한 정보 윤리 의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해하는 것을 목표로 한다[1].

[†] 정 회 원: 한국교원대학교 컴퓨터교육과 박사과정

^{††} 중 신 회 원: 한국교원대학교 컴퓨터교육과 교수(교신저자)

논문접수: 2020년 01월 06일, 심사완료: 2020년 03월 11일, 게재확정: 2020년 03월 16일

이러한 시대적 요구와 관련 교육과정에 의해 소프트웨어교육은 전 학령기 학생들을 대상으로 시행되고 있으며 이러한 흐름은 미국, 유럽 등 주요 선진국뿐 아니라 중국, 인도, 아프리카 등 세계 여러 나라에서도 점차 확대, 보편화 되어가고 있다[2][3]. 이와 관련하여 국내 교육과정에서도 지도 시기 및 내용의 확대에 관한 논의가 진행되는 등 소프트웨어교육은 앞으로 사회 전반에 필수적인 교육내용으로 자리 잡게 될 것으로 생각된다.

현재 우리나라에서도 다양한 교육내용과 교구가 연구, 적용되며 유의미한 효과를 보이고 있다. 한국교육학술정보원에서 진행한 안성훈 외(2017), 양재명 외(2017), 서석민 외(2019)의 연구를 비롯하여 여러 선행 연구에서 소프트웨어교육을 통해 학생들에게 컴퓨팅사고력의 향상이 있음을 밝히고 있다[4][5][6].

하지만 현재 이루어지고 있는 교육내용들을 자세히 살펴볼 때 교육 또는 연구의 관점에서는 가치를 인정할 수 있으나, 학급 또는 학교 등 특정 집단을 기준으로 하고 있다는 한계를 지니고 있다. 노지예(2017)는 연구에서 학습자의 수준을 기준으로 볼 때 컴퓨팅사고력 향상도는 상위집단의 경우 하위집단보다 약하거나 의미가 없다고 분석하였고 안장수(2018)는 연구동향분석을 통하여 초등학생을 대상으로 실시된 연구는 다수 있으나 교육과정에 명시된 교육목표 수준과 연결할 수 있는 연구는 희박하여 실용적이지 못하다고 비판하며 다양한 대상을 바탕으로 한 연구 및 평가도구 개발의 필요성을 제안하였다 [7][8]. 또한 서석민 외(2018)의 분석연구에서도 검사지를 통한 컴퓨팅사고력 기반 문제해결력 향상도는 학교급에 따라 차이가 있으며, 이 중 초등학교가 가장 높지만, 성별 등 특정 기준에서는 차이가 없는 것으로 나타났다[6].

선행 연구사례들에서 볼 수 있듯, 보편적 교육이라는 공교육의 가치에서 생각해 볼 때 특정 학습자만을 만족시키는 교육은 적절하다고 할 수 없으므로 학교와 학급 단위의 집단보다 학생 개인에 더 주목할 필요가 있다고 생각된다.

이에 본 연구는 초등학생 학습자를 보다 작은 단위의 집단으로 구분하여 사용된 교구나 학습자의 수준 등 세부 특성에 따라 컴퓨팅사고력 향상도에 어떠한 차이가 있는지 살펴보고, 초등학생에게 적합한 소프트웨어교육 방법을 모색하여 보고자 한다.

2. 이론적 배경

2.1 교수·학습 활동모형

정찬주 외(1998)는 교수·학습 활동을 특정학습과제의

성공적인 학습을 위해 의도적으로 조성된 조건 아래에서 교사와 학생이 상호작용하는 활동으로 정의하였다[9]. 주제와 활동방법에 따라 다양한 교수·학습활동 모형이 존재하며, 소프트웨어교육에서는 PBL 학습법을 바탕으로 한 Treffinger, Isaksen, Dorvald(2000)의 CPS 모형, 전용주(2017)의 CT-CPS 교수·학습모형과 여러 연구자들의 협업 결과인 김진숙 외(2015)의 5가지 SW교육 교수·학습모델 등이 있으며, 주요 활동절차는 <표 1>과 같다[10][11][12].

표 1. 소프트웨어교육 교수·학습모형(모델)과 활동절차

개발자	교수·학습모형(모델)	활동절차
Treffinger 외	CPS 모형	혼란발견-자료발견-문제발견-아이디어발견-해결안발견-수용안발견
전용주	CT-CPS 모형	문제인식 및 분석-아이디어 구성-설계-구현 및 평가
김진숙 외	DMM 모델	시연 - 모방 - 제작
	UMC 모델	놀이 - 수정 - 재구성
	DDD 모델	탐구 - 설계 - 개발
	NIDS 모델	요구분석-디자인-구현-공유
	DPAA 모델	분해-패턴인식-추상화-알고리즘-프로그래밍

2.2 소프트웨어교육의 주요 교구

교수·학습활동은 각 차시별로 학습목표와 교수·학습방법에 따라 활동지, 교구, 전자매체 등 여러 보조도구를 사용하여 진행된다. 이들은 특성과 목적에 따라 교구, 교수·학습도구 등 다양하게 서술된다[13]. 때문에 본 연구에서는 보조도구의 다양성과 사용목적에 고려하여 교수·학습 활동에 활용된 일련의 보조도구들에 대하여 ‘교구’라는 용어로 정의하였다.

현재 초등학교에서 가장 많이 사용되는 소프트웨어교육의 주요 교구는 다음을 들 수 있다.

2.2.1 언플러그드

컴퓨터 없는 컴퓨팅 교육을 한다는 내용으로 뉴질랜드 Tim Bell 교수의 교육법에서 시작되었다[14]. 보드게임, 활동지 등 다양한 형태의 학습활동을 구성하여 논리적, 활동적인 방법으로 관련 내용을 학습하도록 한다.

2.2.2 비주얼 프로그래밍

교육용 프로그래밍 언어(EPL)를 이용한 프로그래밍 학습법으로, 사용하는 EPL에 따라 블록형과 텍스트형으로 나누어진다. MIT에서 개발한 스크래치(Scratch)와 파

이썬(Python)이 대표적이며, 현재 국내에서는 엔트리와 파이썬 외 다양한 프로그래밍 언어가 사용되고 있고, 2015 개정 교육과정은 성취기준을 통해 초등학생들을 대상으로 블록형 프로그래밍을 통한 교육을 제안하고 있다[15].

2.2.3 피지컬 컴퓨팅

비주얼 프로그래밍을 실제 작동교구에 접목함으로써 실질적인 프로그래밍 결과를 시각, 청각 등 인간의 오감을 통하여 이해할 수 있게 설계된 교육방법이다. 2004년 뉴욕대학교의 Dan O'Sullivan과 Tom Igoe 교수가 인터랙티브 피지컬 시스템(interactive physical systems)을 가르치기 위해 사용한 이후 국내외 여러 연구에서 교육효과가 증명되고 있으며 교구의 특성에 따라 조립형, 완성형 등 다양한 형태의 교구들이 개발, 활용되며 기초교육을 바탕으로 메이커(Maker) 교육으로 이어지고 있다 [16][17].

2.2.4 혼합지도

각각의 교구들이 가지고 있는 장단점을 참고하여 학습 내용이나 위계, 학습자 수준에 따라 효과적인 학습목표 도달을 위해 특정 교수-학습활동에서 여러 교구들을 혼합하여 적용하는 것을 뜻한다[17]. 실제 시범학교 등의 운영사례를 살펴볼 경우 단원의 지도간 강의식 방식에서 PBL형태로 차시간 혼합지도가 이루어지거나 단일 차시 내에서 언플러그드-비주얼프로그래밍 식의 혼합지도가 이루어지는 구성을 볼 수 있다.

2.3 컴퓨팅사고력

소프트웨어교육의 목표는 컴퓨팅사고력의 향상에 있다. 컴퓨팅사고력의 구성 요소는 <표 2>와 같이 Wing의

표 2. 컴퓨팅사고력의 구성요소

Wing(2008)	CSTA&ISTE(2011)	Google for education(2015)
추상화	자료수집	
	자료분석	자료분석
		패턴인식
	자료제시	
	문제분해	분해
	추상화	추상화
패턴 일반화		
알고리즘 및 절차	알고리즘 디자인	
자동화	자동화	
	병렬화	
	시뮬레이션(모의실행)	

구분(2008)과 더불어 CSTA&ISTE와 Google의 두 가지 기준이 주로 인용된다[18][19][20].

다만 교육부(2015)는 운영지침과 교육과정 해설에서 초등학교 수준에서는 문법과 프로그래밍 요소를 최소화하고 체험과 이해중심에 맞출 수 있도록 안내하고 있다 [1][15]. 이를 바탕으로 초등학교에서는 소프트웨어교육에서 학습자의 흥미와 이해를 높일 수 있도록 <표 2>와 같은 절차적인 부분이나 컴퓨터 및 각종 프로그램 등 도구적인 제약을 최소화할 수 있는 여러 가지 방법을 통해 교수-학습 활동과 평가가 이루어지고 있다.

2.4 컴퓨팅사고력 평가도구

특정 영역의 수준을 평가하기 위해서는 평가도구가 필요하다. 이 과정에서 중요한 것은 평가도구의 공인된 타당도와 신뢰도 그리고 객관도이다[21].

2.4.1 기존 평가도구의 한계

컴퓨팅사고력 평가는 대표적으로 KERIS가 연구-시범학교의 평가를 위해 개발한 평가도구가 있다[4][5][6]. 하지만 이는 일부 시범학교의 운영사례만을 평가한 것이기에 때문에 일반 학교에 적용하기 어렵다는 점이 문제로 지적된다. 아울러 개별 연구자들이 사용한 독자적 평가 자료의 경우 개별 상황에는 적합할지 모르나 일반 지도 상황에 적용하기에는 타당도와 신뢰도에서 무리가 있다. 즉 보편적 교육으로서의 컴퓨팅사고력을 평가하기 위해서는 교사들의 교육활동 목적에 따른 평가가 가장 바람직하나, 평가의 객관성을 위해 영국의 사례와 같이 교과 차원에서 학생들에게 적용 가능한 최소한의 개념, 수준, 용어 등의 합의가 있거나 일반화된 검사가 가능한 표준화 검사의 장점이 대안이 될 수 있을 것이다[21][22]. 하지만 연구가 진행되는 시점에서 관련 논의는 아직 진행 중인 것으로 보인다.

2.4.2 평가도구로서의 비버챌린지

이처럼 일정수준 이상의 타당도와 신뢰도를 확보하고 보편적인 공교육사례에 적용할 수 있는 합의 또는 일반화된 검사의 관점에서 본 연구에서는 비버챌린지 문항을 통한 컴퓨팅사고력 평가를 진행하였다. 비버챌린지는 검사도구의 측면에서 국내 교육내용만을 목적으로 한 것이 아니며, 컴퓨팅사고력과의 연관성에 대해 긍/부정적인 여러 평가가 있지만, 국제 학술대회를 통해 문항을 확정하는 절차를 지니고 있어 전수진 외(2018)의 분석에서 볼

수 있듯이 문항의 구성과 내용의 측면에서 국제적 전문가들의 협의와 심사에 따른 타당도와 신뢰도를 확보하고 있다[23]. 또한 <표 3>과 같이 컴퓨팅사고력에 관련된 다양한 영역을 바탕으로 여러 난이도의 문제를 구성하고 있어, 학습자의 수준에 따라 적절한 평가문항 활용이 가능하다. 아울러 전용주 외(2018)는 연구를 통해 비버첼린지의 다수 문항들은 초등학교 소프트웨어교육과정의 성취기준을 만족하는 것으로 나타났다는 점과 공식대회를 통해 학습자 수준을 판단할 수 있는 기준척도를 제공하는 점은 본 연구에서 측정하기 위한 일반학습자의 수준과 향상도 등을 가늠하기에 가장 적합하다고 판단되었다[24].

표 3. 비버첼린지 문항영역과 주요 요소

영역	주요 요소
ALP	알고리즘과 프로그래밍
DSR	자료, 자료구조와 표현
CPH	컴퓨터 처리와 하드웨어
COM	통신과 네트워킹
ISS	상호작용, 시스템과 사회

3. 연구 방법

3.1 연구 대상

본 연구는 초등학교 6학년 5개 학급 학생 116명을 대상으로 진행하였다. 이 중 개인사정으로 활동에 불참하였거나 불성실응답 등으로 인해 분석이 어려운 인원을 제외한 104명을 5개의 그룹(G1부터 G5까지)으로 구성하고, 비버첼린지 2017 공식대회의 성취도를 기준으로 각 그룹을 상위 수준(‘이상’)과 하위 수준(‘이하’)의 세부집

단으로 나누었다. 연구 집단의 구성은 <표 4>와 같다.

표 4. 연구 집단의 구성

그룹	G1	G2	G3	G4	G5
N	20	23	20	21	20
이상	9	8	10	11	15
이하	11	15	10	10	5

3.2 연구 가설

본 연구에서 설정한 연구 가설은 다음과 같다.

- 가. 초등학생의 컴퓨팅사고력 향상도는 교구에 따라 차이가 있을 것이다.
- 나. 동일한 교구를 적용한 상태에서 초등학생들의 컴퓨팅사고력 향상도는 컴퓨팅 사고력에 대한 학습자 수준에 따라 차이가 있을 것이다.

3.3 연구 내용

본 연구에서는 현재 사용되고 있는 교과서의 구성과 내용을 토대로 4가지 소주제로 재구성한 뒤, 5개 집단에 동일한 학습 목표와 교수-학습모형(모델)을 적용하되 교구가 다른 5종류의 교수-학습과정안(14차시)을 구성하여 7주 동안 진행하였다. 적용된 교구는 선도학교 및 시범학교의 운영방식과 교사연수 등에서 사용되는 주요 방법을 참고하여 5종류의 교구를 선정하였다.

소주제별 학습목표와 교수-학습모형은 <표 5>와 같이 주제별 동일한 모형(모델)로 구성하였으며 집단별 활용 교구는 <표 6>, 각 차시별 세부적인 교수-학습활동 내용은 <표 7>과 같다.

표 5. 주제별 학습목표와 교수-학습활동 모형(모델)

구분	SW의 이용	절차적 사고	프로그래밍의 원리	문제해결
학습 목표	생활 속 SW의 사례를 통해 SW 교육의 필요성을 알 수 있다.	절차적 사고의 필요성을 이해하며 이를 문제 해결에 적용할 수 있다.	주요 프로그래밍 원리를 이해하고 절차적 방식을 통하여 문제를 해결할 수 있다.	학습내용을 토대로 주어진 문제를 해결할 수 있다.
교수학습 활동모형 (모델)	DMM 모델	DMM 및 UMC 모델	DMM 및 UMC 모델	창의적 문제해결 (CPS)

표 6. 연구 집단별 교구

그룹	G1	G2	G3	G4	G5
지도방법	언플러그드	비주얼 프로그래밍	피지컬 컴퓨팅	언플러그드 및 비주얼 프로그래밍	언플러그드 및 피지컬 컴퓨팅
사용교구	게임활동 및 활동지	엔트리	햄스터	G1,G2의 방식 혼합	G1,G3의 방식 혼합

표 7. 집단별 교수 학습활동 내용

그룹		G1	G2	G3	G4	G5
주제	차시	교수 학습 활동 내용				
SW의 이용	1~2	SW가 실생활에 이용되는 예 SW를 배워야 하는 필요성				
절차적 사고	3~4	<ul style="list-style-type: none"> 절차적 사고의 개념과 필요성 생활속 문제를 통한 적용 	<ul style="list-style-type: none"> 절차적 사고의 개념과 필요성 엔트리 실습예제를 통한 적용 예 	<ul style="list-style-type: none"> 절차적 사고의 개념과 필요성 햄스터를 이용한 절차적 사고의 적용 예 	<ul style="list-style-type: none"> 절차적 사고의 개념과 필요성 생활속 문제 및 엔트리 실습예제를 이용한 적용 예 	<ul style="list-style-type: none"> 절차적 사고의 개념과 필요성 생활속 문제 및 햄스터를 이용한 적용 예
프로그래밍의 원리	5~6	<ul style="list-style-type: none"> <순차모드> 개념 지도 스택버거 게임 오조봇과 오조코드를 통한 목적지 찾아가기 	<ul style="list-style-type: none"> <순차모드> 개념 지도 엔트리를 통한 원리와 구조 설명 실습과 미션해결 	<ul style="list-style-type: none"> <순차모드> 개념 지도 햄스터를 통한 원리와 구조 설명 실습과 미션해결 	<ul style="list-style-type: none"> <순차모드> 개념 지도 스택버거 게임 엔트리 실습과 미션해결 	<ul style="list-style-type: none"> <순차모드> 개념 지도 스택버거 게임 햄스터 실습과 미션해결
	7~8	<ul style="list-style-type: none"> <반복모드> 개념지도 시그널 보드게임 빙고판 이동하기를 통한 반복블럭의 적용사례 익히기 	<ul style="list-style-type: none"> <반복모드> 개념지도 엔트리를 통한 원리와 구조 설명 실습과 미션해결 	<ul style="list-style-type: none"> <반복모드> 개념 지도 햄스터를 통한 원리와 구조 설명 실습과 미션해결 	<ul style="list-style-type: none"> <반복모드> 개념 지도 시그널 보드게임 엔트리 실습과 미션해결 	<ul style="list-style-type: none"> <반복모드> 개념 지도 시그널 보드게임 햄스터 실습과 미션해결
	9~10	<ul style="list-style-type: none"> <조건모드> 개념지도 엔트리폭탄대소동 조건모드가 적용된 생활속 기기활용 역할극 	<ul style="list-style-type: none"> <조건모드> 개념지도 엔트리를 통한 원리와 구조 설명 실습과 미션해결 	<ul style="list-style-type: none"> <조건모드> 개념 지도 햄스터를 통한 원리와 구조 설명 실습과 미션해결 	<ul style="list-style-type: none"> <조건모드> 개념 지도 엔트리 폭탄대소동 엔트리 실습과 미션해결 	<ul style="list-style-type: none"> <조건모드> 개념 지도 엔트리 폭탄대소동 햄스터 실습과 미션해결
문제해결	11~12	<ul style="list-style-type: none"> 문제해결방법에 대한 복습 생활속 기기에 대한 알고리즘 분석 	<ul style="list-style-type: none"> 문제해결방법에 대한 복습 엔트리를 이용한 게임 시나리오 제작 	<ul style="list-style-type: none"> 문제해결방법에 대한 복습 햄스터 동작에 대한 시나리오 제작 	<ul style="list-style-type: none"> 문제해결방법에 대한 복습 엔트리를 이용한 게임 시나리오 제작 	<ul style="list-style-type: none"> 문제해결방법에 대한 복습 햄스터 동작에 대한 시나리오 제작
	13~14	시나리오 구원 및 보완, 발표				

3.4 평가 도구와 방법

본 연구에서 사용한 평가도구는 2017년 비버챌린지 문항 중 학습자 수준을 감안하여 Group III 문항을 사용하였으며, ALP 영역의 8 문항, DSR 영역의 6 문항, COM 영역의 1 문항, 총 15문항으로 구성되어 있다. 전수진 외 (2018)의 연구에 따르면 본 문항의 신뢰도는 Cronbach α 값이 .716으로 바람직한 수준으로 제시된 바 있으며, 연구 진행간 측정된 Cronbach α 값 또한 .671로 수용 가능한 수준으로 나타났다.

채점은 <표 8>과 같이 공식적으로 정해진 비버챌린지의 채점방법에 따라 실시하여 객관도를 만족시키고자 하였다[24].

표 8. 비버챌린지 문항 채점방법

응답결과 난이도	정답	오답	무응답
쉬움	+6	-2	0
보통	+9	-3	0
어려움	+12	-4	0

각 집단은 본격적인 교수 학습 활동 전 사전 검사를 진행한 후 일원배치분산분석(ANOVA)을 통해 동질집단임을 확인하였다. 7주간 14차시의 교수 학습 활동을 진행 후 사전검사 문항과 동일문항을 이용한 사후검사를 실시하여 사전검사 결과와 비교하는 ANOVA를 통하여 집단간 향상도를 분석하였다. 아울러 비버챌린지 공식대회의 기준척도를 바탕으로 성취도에 따른 ‘학습자 수준’ 요소를 적용하여 G1부터 G5의 5개 집단 각각을 이상/이하의 세부집단으로 재조직하고 세부 집단별 대응집단 t 검증을 통해 사전-사후검사의 성취도 및 영역별 향상도를 분석하여 학습자 특성(수준)에 따른 향상도를 도출한 후, 공분산분석(ANCOVA)을 통하여 각 요인들의 영향력 정도를 비교하였다.

이러한 동형검사 방식을 적용하게 된 것은 검사신뢰도의 한계에도 불구하고 연구가 적용되던 시점은 해당 문항이 국내에 적용된지 3년 정도밖에 되지 않아 검토가능한 문항의 수가 한정되어 있었다는 점과, 국내 교육과정만을 배경으로 하지 않는 비버챌린지의 특성상 매년 다른 문제가 출제되어 동일 그룹의 문제라고 하더라도 다

른 배경지식을 필요로 하거나 문항의 영역 구성이 달라 지기에 동일 난이도를 가지고 있다는 확신을 가질 수 없었기 때문이다. 다만 검사의 한계점을 고려하여 사전-사 후 검사간 문항의 구성 순서를 달리 하거나 7주라는 교육기간을 적용하여 기억에 따른 부정적 영향을 줄이고자 하는 등 검사신뢰도에 영향을 줄 수 있는 요소를 최소화 하고자 노력하였다.

4. 연구 결과

4.1 교구에 따른 향상도 비교

교구에 따라 교수-학습 활동 전과 후의 각 집단 성취도의 변화를 ANOVA를 통하여 검증한 결과는 <표 9>와 같다.

표 9. 사전-사후 ANOVA 적용 결과

	M	SD	df1	df2	F	p
사전검사	51.86	25.04	4	99	2.06	.091
사후검사	59.17	26.98			3.50**	.01

사전검사에서 5개의 집단에 대한 F값은 2.06으로 집단 간 차이가 미미한 동질집단으로 분석되었으나 사후검사에서 F값은 3.50으로 $p < .01$ 수준에서 유의미한 차이가 있는 것으로 나타났다. 이는 교구에 따라 학생들의 컴퓨팅 사고력 향상도에 차이가 발생한다는 것으로 해석되며 첫 번째 가설을 만족한다.

4.2 학습자 수준에 따른 향상도 비교

비버챌린지 2017 공식대회의 성취도를 기준척도로 적용한 ‘이상’/‘이하’ 집단에 대한 대응표본 t 검증 분석결과는 <표 10>과 같다.

표 10. 학습자 수준별 향상도

집단	N	사전		사후		t	p	
		M	SD	M	SD			
G1	이상	9	71.11	11.49	63.00	17.59	1.12	.30
	이하	11	27.27	16.11	50.64	36.86	-2.40	.04*
G2	이상	8	65.00	15.72	81.88	19.17	-1.45	.19
	이하	15	35.67	13.87	50.93	17.01	-3.02	.01**
G3	이상	10	71.10	14.20	63.00	21.53	1.46	.18
	이하	10	30.40	14.78	37.70	26.80	-1.13	.29
G4	이상	11	72.36	15.90	57.91	25.49	2.03	.07
	이하	10	28.00	10.71	43.60	20.98	-2.71	.02*
G5	이상	15	76.60	14.40	82.80	22.01	-0.91	.38
	이하	5	31.60	13.83	57.80	26.35	-1.90	.13

피지컬 컴퓨팅을 적용한 G3와 혼합지도법(언플러그드+피지컬 컴퓨팅)을 적용한 G5 집단은 모두 유의미한 변화를 보이지 않았다. 반면 나머지 세 집단의 경우, 즉 G1(언플러그드)의 ‘이하’ 집단의 p값은 .04, G2(비주얼 프로그래밍)의 ‘이하’ 집단의 p값은 .01, G4(언플러그드+비주얼 프로그래밍)의 ‘이하’ 집단의 p값은 .02로 G1, G2, G4의 ‘이하’ 집단 모두는 $p < .05$ 수준에서 유의미한 향상을 보인 것으로 나타났다. 하지만 ‘이상’ 집단의 경우 유의미한 결과가 나오지 않아 ‘이하’ 집단과 비교할 때 컴퓨팅사고력의 향상이 이루어지지 않았다고 분석되었다. 이러한 결과는 노지예(2018)의 연구결과와 같은 맥락이며, 동일한 교구라도 학습자 특성에 따라 향상도가 다를 수 있다는 두 번째 가설을 만족한다.

4.3 비버챌린지 문항 영역별 검증

컴퓨팅사고력 향상에 관한 보다 자세한 검증을 위하여 비버챌린지 문항에 기반한 영역별 성취도를 검증하였다. 본 연구에서 검증에 활용한 비버챌린지 평가문항은 ALP, DSR, COM의 세 영역 총 16문항으로 구성되어 있으나 1문항으로 구성된 COM 영역은 해당 문항만으로 평가하기에는 객관성이 부족하다고 판단되어 제외한 후 나머지 ALP와 DSR 두 영역에 대해 대응집단 t-검정 방식으로 분석을 진행하였다.

4.3.1 집단별 향상도

각 집단의 영역별 향상도는 <표 11>과 같이 나타났다.

표 11. 집단별 향상도

집단	영역	N	사전		사후		t	p
			M	SD	M	SD		
G1	ALP	20	24.70	16.06	19.00	19.70	1.05	.31
	DSR		18.30	15.30	34.40	15.91	-5.06	.000***
G2	ALP	23	23.78	14.13	28.61	17.67	-1.10	.28
	DSR		18.70	12.42	29.17	9.40	-4.64	.000***
G3	ALP	20	17.75	15.41	17.35	18.44	.13	.90
	DSR		28.60	14.59	30.10	11.47	-0.47	.65
G4	ALP	21	25.67	17.90	17.00	15.87	1.67	.11
	DSR		21.86	14.65	30.76	14.35	-2.64	.02*
G5	ALP	20	28.30	12.34	36.70	16.96	-1.88	.08
	DSR		32.15	17.07	36.55	13.16	-0.90	.38

5개의 집단(G1~G5)에서 모두 DSR 영역의 향상이 관찰되었으며, 이 중 G1(언플러그드)과, G2(비주얼 프로그래밍)의 p값은 .000, G4(언플러그드+비주얼프로그래밍)의 p값은 .02로 각각 $p < .05$ 이상의 수준에서 유의미한 향

상이 나타난 것으로 분석되었으나 ALP 영역은 전체 집단에서 유의미한 향상이 나타나지 않았다.

4.3.2 학습자 수준별 향상도

학습자 수준에 따른 영역별 향상도는 <표 12>와 같이 분석되었다.

표 12. 학습자 수준별 향상도

집단	영역	N	사전		사후		t	p	
			M	SD	M	SD			
G1	이상	ALP	9	34.78	12.41	18.44	17.87	2.10	.07
			DSR	30.33	9.49	41.22	8.74	-2.54	.04*
	이하	ALP	11	16.45	14.16	19.45	21.93	-4.45	.67
			DSR	8.45	11.67	28.81	18.54	-4.68	.001***
G2	이상	ALP	8	32.50	15.34	41.88	13.43	-1.12	.30
			DSR	28.50	10.01	35.00	7.71	-1.24	.25
	이하	ALP	15	19.13	11.38	21.53	15.70	-.47	.65
			DSR	13.47	10.40	26.07	8.92	-6.25	.000***
G3	이상	ALP	10	28.50	9.98	26.20	17.54	.49	.64
			DSR	37.40	12.54	33.80	8.01	.82	.43
	이하	ALP	10	7.00	12.04	8.50	15.37	-.38	.71
			DSR	19.80	10.96	26.40	13.53	-1.54	.16
G4	이상	ALP	11	36.45	12.78	19.91	17.80	2.60	.03
			DSR	30.64	10.35	34.91	14.37	-.89	.39
	이하	ALP	10	13.80	15.24	13.80	13.62	.00	1.00
			DSR	12.20	12.66	26.20	13.57	-3.19	.01**
G5	이상	ALP	15	32.40	10.28	41.73	14.87	-1.70	.11
			DSR	38.73	10.42	37.07	12.94	.39	.70
	이하	ALP	5	16.00	10.05	21.60	14.57	.71	.52
			DSR	12.40	18.86	35.00	15.23	-1.82	.14

학습자의 성취수준을 바탕으로 본 결과 또한 ALP 영역은 대부분의 집단에서 유의미한 향상이 나타나지 않는 반면, DSR 영역은 G1(언플러그드) ‘이하’ 집단의 p값은 .001, G2(비주얼 프로그래밍)의 ‘이하’ 집단의 p값은 .000, G4(언플러그드+비주얼 프로그래밍)의 ‘이하’ 집단 p값은 .01로 p<.01 이상의 유의미한 향상을 나타내었다. ‘이상’ 집단의 경우 ALP와 DSR 대부분의 영역에서 유의미한 향상을 보이지 않았는데, 이는 현 교과서의 구성이 알고리즘에 대한 지식적 내용보다 소프트웨어의 이해와 체험에 중점을 맞추고 있다는 점에 기인하는 것으로 보인다. 즉, ‘이상’집단이 가지고 있는 지식수준과 내용에 비해 교수·학습활동의 내용과 수준이 적절치 않은 것으로 해석된다.

지금까지 분석한 내용을 종합하면 현재의 교과서 내용을 바탕으로 한 초등학교의 소프트웨어교육은 언플러그드와 비주얼 프로그래밍을 위주로 한 지도 방법으로 교수·학습활동이 진행될 경우, 선행지식이 낮은 학습자 집

단에서 DSR(자료 및 자료구조) 영역이 가장 많은 향상을 보이나, 일정 수준 이상의 학습자에게는 별다른 영향을 미치지 않는 것으로 요약된다.

4.3.3 요인별 영향력 비교

지금까지 제시한 바와 같이 교수·학습 활동 이후 5 종류의 집단(G1~G5)의 컴퓨팅사고력 향상도는 집단 간 차이가 있었으며 특히 이런 차이의 원인으로서는 교구와 학습자 수준이 주요 원인으로 작용한 것으로 파악되었다. 해당 요인들의 영향력을 파악하기 위하여 사전 성취도를 공변량으로 설정한 후 사후 성취도에 대한 공분산분석을 실시하여 각 요인들의 효과크기를 측정하였다.

효과크기(η^2)에 대한 Cohen의 해석을 기준으로 볼 때 <표 13>과 같이 교구 .129 와 학습자 수준 .082로 두 요인 모두 중간 이상의 영향을 미치는 것으로 나타났으며 상대적으로 교구 요인이 더 큰 영향을 미치는 것으로 분석되었다.

이를 본 연구의 앞선 분석 내용과 비교하여 볼 때 효과적인 소프트웨어교육을 위해서는 교수·학습 방법이 가장 먼저 고려되어야 하겠으나, 그 외 학습자 수준 등 다양한 요인을 고려해야 더욱 질 높은 교육효과를 이룰 수 있다는 것으로 해석할 수 있다.

표 13. 공분산분석 결과

소스	제곱합	자유도	평균제곱	F	p	η^2
수정된 모형	32429.103a	20	1621.455	3.161	0.000	0.432
절편	25781.673	1	25781.673	50.265	0.000	0.377
사전성취	3667.293	1	3667.293	7.150	0.009	0.079
학습자 수준	3803.606	1	3803.606	7.416	0.008	0.082
교구	6322.301	4	1580.575	3.082	0.020	0.129
학습자 수준 * 교구	858.942	4	214.736	0.419	0.795	0.020
오차	42571.782	83	512.913			
전체	439152.000	104				
수정 합계	75000.885	103				

a. R 제곱 = .432 (수정된 R 제곱 = .296)

5. 결론 및 제언

5.1 결론

본 연구를 통하여 다음과 같은 결론을 도출할 수 있다. 첫째, 소프트웨어교육 활동에 따른 컴퓨팅사고력의 변화에 있어 학생들의 사전지식 등 학습자 수준에 따라 컴

퓨팅사고력의 향상도 차이가 큰 것으로 나타났고 현재의 주요 교구는 학습자 수준이 낮은 학생들에게 상대적으로 높은 효과가 있는 것으로 나타났다. 집단별 사례수가 낮은 한계로 인하여 일반화에는 한계가 있으나, 본 결과를 통해 학급위주의 활동보다는 집단이나 학습자의 사전수준 등 개별적 특성을 고려한 다양한 활동방식을 적용할 필요성이 제기된다. 특히 상대적으로 높은 수준의 지식과 능력을 가지고 있는 학생들을 위한 적절한 교수·학습 활동과 교구에 대한 연구가 필요하다.

둘째, 학생들의 향상도는 DSR 영역에서 유의미한 향상을 보였으며, 이는 구글 및 CSTA&ISTE의 컴퓨팅사고력 요소와 비교하여 볼 때도 추상화 영역에 해당한다. 소프트웨어교육을 시작하는 학습자의 특성을 잘 나타내는 것으로 초등학생 수준의 성취기준과도 부합하므로 후속 연구나 향후 지도내용구성에 있어 프로그래밍적 요소에 비해 문제해결을 위한 개념, 절차이해에 주목한 지도내용 구성이 초등학생에게는 적합한 것으로 생각된다.

셋째, 교구에 있어 언플러그드와 비주얼프로그래밍, 이들을 복합적으로 적용한 혼합지도법이 상대적으로 높은 효과를 보이는 것으로 나타났다. 이는 일반 초등학교의 소프트웨어교육에 있어서 고려할 중요한 시사점이라고 할 수 있다.

5.2 제언

본 연구의 결과해석에 있어 다음과 같은 점에서 주의가 필요하다.

첫째, 집단을 세분화함으로써 인해 각 집단이 소수인원으로 구성되었다는 점과, 7주라는 비교적 긴 기간을 두고 연구를 진행하였으나, 동일문항을 바탕으로 사전-사후 조사가 이루어졌다는 점에서 연구결과를 전체집단으로 일반화하기에는 한계가 있으므로 활용시 주의가 요구된다.

둘째, 본 연구에서 도출한 교수·학습 활동은 교과서에 제시된 내용을 바탕으로 지도내용을 재구성하여 일정기간 진행한 것이므로, 연구진행간 성취도에 영향을 미친 다른 요인을 완전히 배제하기는 어렵다.

셋째, 본 연구에서 사용한 비버첼린지 문항을 이용한 평가와 분석은 2017년도 문항과 점수기준을 인용하여 진행하였으며, 앞서 분석한 바와 같이 평가도구로서의 기준에 상당히 부합한다고 여겨지나, 일부 문항의 경우 국내 교육과정과 상이한 배경지식을 바탕으로 구성되어 있거나 매년 다른 문항이 출제되므로 평가도구로 활용 시 동형의 다른 문항을 이용하는 등 연구자의 보완이 필요하다.

하지만 본 연구는 실제 적용중인 교과서를 바탕으로

다양한 교구를 사용한 교수·학습 활동을 다수 집단에 적용 후 복합적으로 분석하였다는 점을 통해 공교육에서 소프트웨어교육을 어떤 방식으로 접근해야 하는가에 대한 현실적인 방안을 제언하였다는 것에 의의가 있다. 향후 다양한 후속 연구를 통하여 초등학생을 위한 효율적인 소프트웨어교육의 교수·학습 활동 구성을 위한 출발점이 될 수 있기를 기대한다.

참고문헌

- [1] 교육부 (2015). **소프트웨어교육 운영 지침**. 서울:교육부
- [2] 김현철 (2017). **정책연구-2017-01호 4차 산업혁명에 대비한 SW 융합인재 양성방안**(2017). 국가과학기술자문회의.
- [3] 김민석 (2018). **이슈리포트 2018-53호 SW인재 양성을 위한 국내의 SW교육 현황 및 시사점**.
- [4] 안성훈·김종민·송태욱·이철현·김성현·이정환·고수빈·박형용 (2017). **연구보고 KR 2016-4. 2016년도 SW 교육 연구학교 효과성 분석 연구**. 한국교육학술정보원
- [5] 양재명·이원규·김자미·윤일규·서정희·우호성·양혜지·김민정·최희정 (2017). **연구보고 CR 2017-11. 2017년도 소프트웨어(SW)교육 연구학교 현황 및 효과성 분석**. 한국교육학술정보원
- [6] 서석민·김수영·강슬기·구찬동·이원민 (2019). **연구보고 KR 2018-1 2018년도 소프트웨어(SW)교육 연구학교 현황 및 효과성 분석**. 한국교육학술정보원
- [7] 노지에 (2017). **로봇활용 SW 교육 프로그램의 효과성 검증**. 박사학위논문. 이화여자대학교.
- [8] 안장수 (2018). **초·중·고 소프트웨어(SW) 교육을 위한 국내 컴퓨팅사고력 연구동향 분석**. 석사학위논문 경상대학교
- [9] 정찬주·최태식·고명석·강영숙 (1998). **교사양성을 위한 교육방법론**. 서울: 학이당.
- [10] Treffinger, D. J., Isaksen, S. G., & Dorval, K. B. (2000). *Creative Problem Solving: An introduction*. (3rd Ed.). Waco, TX: Prufrock Press
- [11] 전용주 (2017). **새로운 교육과정의 소프트웨어교육을 위한 컴퓨팅사고력 기반 창의적 문제해결(CI-CPS) 수업모형의 개발 및 적용**. 박사학위논문. 한국교원대학교
- [12] 김진숙·한선관·김수환·정순원·양재명·장의덕·김정남·류미영·이진태·전수진·김상홍 (2015). **수탁연구 CR 2015-35. SW교육 교수학습 모형 개발 연구**. 한국교육개발원·한국교육학술정보원.
- [13] 신나민·하오선·장연주·박종향 (2019). **이판사판 교육 방법 및 교육공학(2판)**. 서울: 박영Story.
- [14] 김송미·김용·이원규(역) (2010). **놀이로 배우는 컴퓨터 과학**. 서울: 휴먼사이언스.
- [15] 교육부 (2015). **실과(기술·가정) 교육과정**. 서울: 교육부
- [16] 이영재·김영식 (2015) **개정교육과정에 따른 초등학교 소프트웨어교육을 위한 피지컬 컴퓨팅 교구 선택기준 개발 및 적용**. 석사학위논문. 한국교원대학교.

[17] O'Sullivan, D., & Igoe, T. (2004). *Physical computing: sensing and controlling the physical world with computers*. Course Technology Press.

[18] Wing, J. M. (2008). *Computational Thinking and thinking about computing*. Phil. Trans. R. Soc. A, 366 (1881). 3717-3725.

[19] CSTA. K-12 *Computer Science Framework*. <https://k12cs.org>.

[20] Goggle for Education. *Exploring Computational Thinking*. <https://edu.google.com/resources/programs/exploring-computational-thinking/>

[21] 김석우 (2015). **교육평가의 이해(2판)**. 서울:학지사

[22] 마대성·정인기(역) (2014). **초등SW교육 컴퓨팅 기초다지기 교사용 지도서(4단계)**. 서울: 미래창조과학부.한국과학창의재단.

[23] 전수진·전용주·김슬기·김도용·정인기 (2018). 비버챌린지 2017을 통한 초등학생들의 컴퓨팅 사고 수준 분석. **정보교육학회논문지**, 22(3), 345-356.

[24] 전용주·정용열·김인주·김지혜·이현아·김동윤 (2018). 소프트웨어 교육에서 비버챌린지를 활용한 교수·학습 및 평가 방법 탐색. **컴퓨터교육학회논문지**, 21(6), 63-82.



이 영 재

2003년 대구교육대학교 초등교육 (교육학학사)
 2017년 한국교원대학교 컴퓨터교육과 (교육학석사)
 2018년~현재 한국교원대학교 컴퓨터교육과 박사과정
 2003년~현재 초등학교 교사

관심분야: SW교육, 인공지능교육, 정보교육과정, 교수학습방법, 피지컬 컴퓨팅
 e-mail: opaluc@gmail.com



김 영 식

1982년 서울대학교 전기공학과 (공학사)
 1987년 노스캐롤라이나주립대학교 전기 및 컴퓨터공학과 (공학석사)
 1993년 노스캐롤라이나주립대학교 전기 및 컴퓨터공학 (공학박사)
 1993년~1994년 한국전자통신연구소 선임연구원

1994년~현재 한국교원대학교 컴퓨터교육과 교수
 관심분야: 컴퓨터 교육, 프로그래밍 교육, 피지컬 컴퓨팅
 e-mail: kimys@knue.ac.kr