

# 예비교원 컴퓨팅 사고 이해도 분석을 통한 소프트웨어 교육 역량 강화 방안

김 철

광주교육대학교 컴퓨터교육과

## 요약

세계적으로 교육 분야에서 컴퓨팅 사고력 증진을 위해 많은 노력을 기울이고 있다. 본 논문에서는 SW 역량과 컴퓨팅 사고력 관계를 정의한 후, 컴퓨팅 사고력 측정을 통해 예비교원 컴퓨팅 사고력 현황을 파악하고자 한다. 이를 위해 한 학기 동안 엔트리 프로그래밍 언어 수업을 받은 예비교원을 대상으로 SW교육에 대한 태도와 컴퓨팅 사고력에 대한 설문은 실시하였다. 설문 분석 결과 예비교원의 컴퓨팅 사고력 중 추상화, 자료 분석, 자동화에 대한 부분의 개선이 필요함을 보였다. 예비 교원을 교육할 때 SW 개발 도구인 엔트리를 벗어나, 분석과 설계 역량을 키울 수 있는 교육 방법이 필요함을 알 수 있었다.

키워드 : 컴퓨팅 사고력, 소프트웨어 교육, 예비교원, 소프트웨어 능력, 엔트리

## A Study on Strengthening Software Education Capability through Computational Thinking Understanding of pre-service Teachers

Chul Kim

Dept. of Computer Education, Gwangju National University of Education

## Abstract

There is a great deal of effort to improve Computational Thinking in education. In this paper, we define the relationship between SW capability and Computational Thinking, and then examine the current status of pre-service teacher's computing thinking. For this purpose, a survey about the attitude toward SW education and computing thinking was administered to prospective teachers who learned Entry software for one semester. The results of the survey showed that the improvement of abstraction, data analysis, and automation among the pre-service teacher's computing thinking is needed. When educating preliminary teachers, we can see that we need an education method that can increase the analysis and design capability beyond the Entry Software development tool.

Keywords : Computational Thinking, CT, Software Education, Entry, Pre-service Teacher

## 1. 서론

4차 산업혁명 시대에 미래 인재 육성 핵심 영역으로 소프트웨어 교육이 자리 잡고 있다. 영국은 2014년 9월부터 ‘컴퓨팅(Computing)’ 과목을 신설하여 초·중등학교의 모든 학령(5세~16세)에서 필수로 이수해야 하는 과목으로 지정하여 시행하고 있다. 미국은 2016년부터 유치원에서 고등학생까지 컴퓨터과학교육을 위한 ‘Computer Science for All’ 추진하고 있다[23].

우리나라 교육부에서는 전 세계적으로 불고 있는 소프트웨어 교육에 대응하고자 2015개정교육과정에서 소프트웨어 교육을 필수교육과정으로 도입하였다. 소프트웨어 교육은 2015 개정 교육과정이 적용되는 2018년에는 중학교과정에 독립교과로 필수 교육과정으로 반영이 되었고, 2019년부터는 초등학교의 6학년에 제시되어 교육과정 영역에 반영되고 있다[16].

소프트웨어 교육의 핵심은 소프트웨어 개발의 기본적인 개념과 원리를 이해하고 이를 통해 다양한 문제를 해결하는 과정에서 논리적이고 창의적인 사고력을 함양하는 컴퓨팅 사고력(Computational Thinking, CT) 습득이라 할 수 있다. Wing(2006)은 ‘Computational Thinking’를 제안하면서 추상화(Abstraction)와 분해(Decomposition)를 통해 복잡한 문제를 해결하는 것이며, 패턴(Pattern)을 찾고, 알고리즘(Algorithm) 작성을 통해 문제 해결 방법을 습득하는 것이라고 하였다. 그리고 컴퓨팅 사고력의 핵심은 프로그래밍이 아닌 개념화에 있으며, 모든 사람이 갖추어야 하는 핵심 역량으로 인간의 사고 방법이라 하였다[24]. 최형신(2016)은 컴퓨팅 사고력을 알고리즘적 사고, 평가, 분해, 추상화 및 일반화로 설정하였다[3]. 미국의 National Research Council of the National Academies에서는 CT를 문제해결 사고 과정이라 보았으며, 일반적인 문제해결과 다르게 컴퓨팅 파워를 문제해결력에 연결시킬 수 있는 역량이라는 점이라고 보았다[17].

학교 교육 현장에서 소프트웨어 교육의 주체는 교사로, 교사의 지도역량이 중요하다. 조미현(2018)은 소프트웨어 교육의 효과를 이루기 위해서는 교사의 역량과 이해에 의존한다고 보았으며[8], 정인기(2017)는 초등예비교원에게 컴퓨팅사고를 적절하면서 효과적으로 교육을 실시하는 것은 초등학교 SW교육의 성공과 연결되어 있다고 보았다[6]. 오지혜(2017)는 ‘교사의 교수역량이

초등학생의 학업성취도에 미치는 영향에 대한 다층모형 분석’에서 교사의 교수역량 중 교수실행 역량이 학업성취도에 직접적인 영향을 주는 것으로 파악했으며, 교수역량의 간접적인 영향도 있을 수 있음을 고려한다면 실제 영향력은 그 이상일 수도 있다고 보았다[18]. 이희숙(2011)은 교원의 전문성 계발을 위한 다양한 활동들이 학생들의 학업성취도에 미치는 영향을 분석하여 교사간의 차이가 학생들의 성취도에 영향을 줄 수 있음을 분석하였다[15]. 이정민(2019)은 ‘초등예비교원의 SW교육에 대한 인식, 경험의 질적 탐구’를 통해 초등 예비교원의 지도 역량이 성숙되어야 함을 연구하였다[14].

본 논문에서는 예비교원의 컴퓨팅 사고력에 대한 사고력 측정을 통해 예비교원 소프트웨어 교육 프레임워크를 도출하고자 한다. 초등 예비교원 교육을 받고 있는 국내 G 교육대학교 1학년을 대상으로 SW교육 효과성의 인지적 능력 측정을 검사하기 위해 설문을 실시하였다. 설문 분석을 통하여 예비교원의 SW 교육에 대한 인지도와 지도 역량 분석을 통해 컴퓨팅 사고력을 파악하고자 하였다.

## 2. 관련연구

### 2.1 컴퓨팅 사고력 개념 및 구성요소

NRC(2010)에서는 컴퓨팅 사고력을 언어와 프로그래밍의 중요성(Language and the Importance of Programming), 추상화의 자동화(Automation of Abstractions), 인지 도구(Cognitive Tool), 컴퓨터를 프로그래밍하지 않는 문맥상황(Contexts Without Programming a Computer) 등으로 정의했다[17].

Wing(2006)은 ‘컴퓨팅 사고력은 컴퓨터 과학자뿐만 아니라 누구나 배워서 활용할 수 있는 보편적 사고이자 기술이다’라고 했다. 또한 컴퓨팅 사고력의 특징을 다음과 같이 정의했다. 컴퓨팅 사고력의 핵심은 프로그래밍이 아닌 개념화에 있다. 컴퓨팅 사고력은 단순 반복적인 기술이 아닌 모든 사람이 갖추어야 하는 핵심 역량이다. 컴퓨팅 사고력은 컴퓨터가 아닌 인간의 사고 방법이다. 컴퓨팅 사고력은 수학적 사고와 공학적 사고를 보완하고 결합한다. 컴퓨팅 사고력은 인공물이 아닌 아이디어이다. 컴퓨팅 사고력은 모두를 위한 것이다[24].

Denning(2015)은 컴퓨팅 원리와 결합하여 컴퓨팅 사고력의 구성요소로서 알고리즘 및 자료표현, 병렬 처리 능력, 자동화 및 패턴, 자료 분석 및 시뮬레이션, 분해, 추상화 등으로 보았다[4]. ISTE(International Society for in Education)와 CSTA(Computer Science Teacher Association)는 컴퓨팅 사고력의 구성요소로 자료수집, 자료 분석, 자료표현, 문제분해, 추상화, 알고리즘 과 절차, 자동화, 시뮬레이션, 병렬화로 정의하였다[21].

## 2.2 교사의 교수역량이 학업에 미치는 영향 연구

컴퓨팅 사고력을 효과적으로 가르치기 위해서는 교사의 교수 역량이 학업에 미치는 영향을 이해해야 한다. 컴퓨팅 사고력 습득에 있어 교수 역량이 학업 성취에 미치는 영향을 알아볼 필요가 있다.

오지혜(2017)는 ‘교사의 교수역량이 초등학생의 학업 성취도에 미치는 영향에 대한 다층모형 분석’에서 교수 역량을 ‘교수활동과 관련된 맥락에서 가르치는 일을 성공적으로 수행하기 위해 요구되는 교사의 개인적 특성, 태도, 지식, 기술과 같은 자질’이라고 정의 하였다. 교사의 교수역량 중 교수실행 역량이 학업성취도에 직접적인 영향을 주는 것으로 파악했으며, 교수역량의 간접적인 영향도 있을 수 있음을 고려한다면 실제 영향력은 그 이상일 수도 있다고 보았다[18]. 진성희(2009)는 교수 역량을 ‘가르치는 일을 성공적으로 수행하기 위해 요구되는 교사의 자질’이라고 정의하였다[7].

이희숙(2011)은 교원의 전문성 개발을 위한 다양한 활동들이 학생들의 학업성취도에 미치는 영향을 분석하여 교사 간의 차이가 학생들의 성취도에 영향을 줄 수 있음을 분석하였다[15]. 이정민(2019)은 ‘초등예비교원의 SW교육에 대한 인식, 경험의 질적 탐구’를 통해 초등 예비 교원의 지도 역량이 성숙되어야 함을 연구하였다 [14]. 박숙현(2015)의 ‘여중생이 지각한 수학교사의 실천적 교수역량이 교육성취도에 미치는 영향’에서 ‘학생들의 수학 교과에 대한 교육성취도 향상을 위해서는 수학 교사의 실천적 교수역량과 학생들의 수학학습태도가 중요한 역할을 한다.’는 것을 연구하였다[20]. 정인기(2017)는 ‘교육대학교 로봇 활용 소프트웨어 교육 과정 개발을 위한 예비 교사의 인식 조사 연구’에서 로봇을 활용한 소프트웨어 교육이 효과적으로 실시되려면 교사에 대한

교육이 효과적으로 이루어져야 한다고 보았다[6]. 이를 위해 예비 교사를 위한 효과적인 교육과정을 개발하기 위해서는 로봇 활용 소프트웨어 교육을 실시하게 될 예비 교사들의 수준을 파악할 필요하다고 보고 이를 위해 설문 조사 분석을 하였다.

## 2.3 컴퓨팅 사고력 효과성 측정에 대한 연구

교수 역량의 컴퓨팅 사고력을 살펴보기 위해서는 컴퓨팅 사고력 효과성 연구를 살펴볼 필요가 있다. 컴퓨팅 사고력에 대한 효과성 연구를 위해서는 먼저 컴퓨팅 사고력 구성요소의 정의한 후 이에 대해 평가해야 한다.

Brennan(2012)은 설계 기반 학습 활동이 컴퓨팅 사고력의 발달을 지원한다는 것을 가정으로 미디어와 상호작용하는 프로그래밍을 통해 컴퓨팅 사고력을 측정하는 도구를 개발하였다[1]. Brennan은 Computational Thinking(CT)의 영역을 크게 컴퓨팅 개념(computational concept), 컴퓨팅 실천(computational practice), 컴퓨팅 관점(computational perspective) 3개로 나누고, 세부 구성요소들을 설정했다. 컴퓨팅 개념(computational concept)은 sequences, conditionals, loops, operators, parallelism, data, events, 컴퓨팅 실천(computational practice) incremental and iterative, testing and debugging, reusing and remixing, abstracting and modularizing, 컴퓨팅 관점(computational perspective)은 expressing, connecting, questioning으로 세부 요소를 나누었다[1,2].

미국의 국립 과학 재단의 지원을 받아 개발된 컴퓨팅 사고력 평가 연구인 Principled Assessment of Computational Thinking(PACT)은 컴퓨터 과학 학습을 위한 모델 교육과정인 Exploring Computer Science(ECS)를 바탕으로 하는 컴퓨팅 사고력의 학습을 평가하기 위한 프레임워크를 개발하였다[22]. PACT의 Computational Thinking(CT) 구성 요소로는 크게 컴퓨터 과학개념, 연구능력, 비인지 능력으로 구분했다. 컴퓨터 과학 개념을 알고리즘, 프로그래밍, 반복, 추상화, 디버깅과 테스트, 변수로 연구 능력을 평가, 탐색, 분석, 설명, 정교화, 모델로, 비인지적 능력을 의사소통, 협력/팀워크, 리더십, 자아 개념, 집념 등으로 세부 구성요소를 나누었다. PACT는 수업 환경에서 나타날 수 있는 학습자의 반응들을 근거로 하여 컴퓨팅 사고력을 평가하기 위한 프레임워크를 제시하였다[19,22]. SRI(2015) 교육팀은 최근 컴퓨팅 사고력의 능력을 평가하는 유효한 증거를 제시할

수 있도록 평가 과제를 설계하는 원리 즉 증거 기반 디자인 (ECD) 프레임 워크를 사용하여 평가하는 방법을 제시하고 있다[22].

최형신(2017)은 ‘소프트웨어 프로그래밍교육을 통한 컴퓨팅 사고력 개발에 관한 국내 연구 고찰’에서 컴퓨팅 사고력을 평가하기 위해 Brennan(2012) 프레임워크를 기반으로 크게 컴퓨팅 개념, 컴퓨팅 수행, 컴퓨팅 관점으로 나누고, 컴퓨팅 개념을 순차구조, 조건구조, 반복구조, 연산자, 병행처리, 데이터(변수), 이벤트로, 컴퓨팅 수행을 점진적인 시도-개발, 테스트-디버깅, 재사용-재조합, 추상화-모듈화로, 컴퓨팅 관점을 표현하기, 연계하기, 질문하기로 나누었다. 또한 국내 컴퓨팅 사고력 개발 프로그래밍 교수 접근법을 문제 해결적 접근, 협력적 상호작용 접근, 성찰기반 접근, 발견 학습 및 디자인 접근으로 나누어 분석하였다[2].

KERIS(2017)는 ‘2017년도 소프트웨어(SW)교육 효과성 측정도구 개발 연구’를 통해 컴퓨팅 사고력 측정 지표를 개발하였다. 컴퓨팅 사고력을 <Table 1>과 같이 SW관련 역량과 연관 지어 정의하고 ‘컴퓨팅 사고력’과 ‘SW교육에 대한 태도’가 각각 어떻게 변화하였는가를 분석하는 효과성 분석 도구를 개발하였다. 이를 통해 학습자의 컴퓨팅 사고력 관련 역량을 전체적으로 측정하고자 하였다[10].

<Table 1> Computational Thinking

SW-related Capabilities	Computational Thinking
Analysis ability	Data collecting
	Data analysis
	Data Representation
	Problem decomposition
Design ability	Abstraction
	Algorithm
Implementation ability	Automation
Reasoning ability	Simulation
	Parallelism

KOFAC(2015)는 ‘초중등 SW교육 실태조사 및 효과성 측정지표 개발 연구’에서 SW효과성 지표를 크게 정의적 영역과 인지적 영역으로 나누었다. SW교육 효과성 지표 인지적 영역은 Computational Concepts(CC)로 SW 교육에서 컴퓨팅 사고력을 학습하거나 향상시키기 위해 배워야 하는 중요한 개념으로, Computational Practice(CP)는 SW 교육에서 컴퓨팅 사고력을 학습하거나 향상시키기

위한 수행 능력 및 실행과정으로, Computational Material & Output(CMO)는 컴퓨팅 사고력의 기저가 되는 알고리즘, 데이터, 프로그램을 Computational Material로, 컴퓨팅 사고 역량의 결과로 추상화, 자동화를 Computational Output 으로 설정하였다. 그리고 효과성 지표로 <Table 2>와 같이 세부 영역을 나누어 지표를 개발하였다[13].

<Table 2> Effectiveness index of SW education

Domain	Details
Computational Material & Output(CMO)	Algorithm, Data, Program, Automation, Abstraction
Computational Concepts(CC)	Sequential, Iterative, Selection algorithm, List, Variable, Function, Calculate, I/O
Computational Practice(CP)	Analyze, Collect, Represent, Design, Implement, Correct an error, decomposition

박영신(2017)은 컴퓨팅 사고력의 실천을 분석하는데 사용할 컴퓨팅 사고 실천 분석 도구(Computational Thinking\_S TEAM\_Analyzing Tool, CT\_STEAM\_AT)를 개발하였으며, CT\_STEAM\_AT를 통해 STEAM 프로그램에 반영된 CT 종류 및 수준을 파악하는데 노력 하였다.[21]. 주여진(2018)은 컴퓨팅 사고력의 추상화를 객체 추상화, 데이터 추상화, 절차 추상화로 분류하고, 추상화 역량 평가를 위해 추상화 성취기준과 평가기준을 개발하였다[9]. 전수진(2016)은 SW 교육에서 학습자의 컴퓨팅 사고력을 효과적으로 평가하기 위한 방법으로써 실험하기 및 반복하기, 테스트와 디버깅, 재사용과 재구성, 추상화와 모듈화의 4가지 영역으로 서술형 수행평가 도구를 구성하여 평가하였다[5].

김민정(2017)은 컴퓨팅 사고력 측정에 사용되고 있는 도구 분석을 통한 새로운 검사도구 개발방향을 제시하였다[11]. 김성식(2019)는 문제해결 프로그래밍 교육에서 컴퓨팅 사고력 평가를 위한 지필형 검사지 및 자기 보고식 설문지 활용 가능성 연구를 진행하였다[12].

### 3. 예비교원 SW역량과 컴퓨팅사고력 효과성분석

#### 3.1. 컴퓨팅 사고력 효과성 측정 영역

예비교원의 컴퓨팅 사고력 효과성을 측정하기 위해 KERIS(2017)의 ‘2017년도 소프트웨어(SW)교육 효과성 측

정도구 개발 연구'에서 정의한 컴퓨팅 사고력 항목인 <Table 1>을 수정하여 <Table 3>과 같이 SW 관련 역량과 연관 지어 컴퓨팅 사고력 항목을 정의하였다. 본 논문에서는 <Table 3>과 SW역량의 분석 능력에 필요한 컴퓨팅 사고력을 추상화, 자료 분석, 문제 분해로 정의하였고, 설계능력을 패턴 찾기, 알고리즘으로 정의하였다. 그리고 SW역량의 구현 능력과 일반 능력은 KERIS(2017)에서 제안한 것과 동일하게 정의하였다[10].

본 논문에서는 예비교원에 대한 SW 교육의 정의적 영역을 평가하기 위해 16개 문항을 만들었다. 예비교원의 SW교육의 인지적 영역인 '컴퓨팅 사고력'을 평가하기 위해 <Table 3>에 정의된 것 중에 분석능력의 자료 분석, 추상화, 설계능력의 패턴 찾기와 알고리즘, 구현 능력의 자동화 분야의 문항 등 총 7개 문항을 만들었다. 이를 통해 예비교원의 'SW 교육에 대한 태도'와 '컴퓨팅 사고력'을 분석하고자 한다.

<Table 3> Computational Thinking

SW-related Capabilities	Computational thinking	Details
Analysis ability	Abstraction	Remove unnecessary elements, leaving only the key elements needed for troubleshooting-find features.
	Data analysis	Finding data for problem solving and organizing data.
	Problem decomposition	Decomposition Problem Into Small Problems to Solve Problem.
Design ability	Find pattern	Find repeating rules.
	Algorithm	Find a set of sequences to solve a problem.
Implementation ability	Automation	Automating or acting on tasks that require repetitive tasks using a computer or electronic device.
	Testing	Experiment based on modeling and models to represent data or procedures.
General ability	Application and generalization	Using structured models designed to solve individual or special needs simultaneously to fit more scope or general matters.

### 3.2. 설문 대상 및 분석 방법

본 연구에서는 국내 G 교육대학교 1학년 학생들을 대상으로 구글 설문지를 이용하여 설문조사하였다. 설문 조사에 응한 학생들은 교양과목에서 주당 2시간씩 한 학기 동안 엔트리(Entry) 프로그래밍 언어수업을 받았다.

설문 조사는 엔트리 프로그래밍 언어를 수강한 후 이루어졌기 때문에 SW 역량 중 분석 능력에 해당하는 컴퓨팅 사고인 추상화, 자료 분석, 문제 분해를 습득했고, SW 역량 중 설계능력에 해당하는 컴퓨팅 사고력인 패턴 찾기, 알고리즘을 학습했으며, SW 역량 중 구현 능력에 해당하는 컴퓨팅 사고력인 자동화를 이해했다고 볼 수 있다. 본 설문에 참여한 학생들은 138명으로 성별로는 남자 55명, 여자 82명, 무응답 1명이 참여하였다.

설문 분석은 IBM SPSS Statistics 25를 사용하여 분석하였다.

### 3.3. 예비 교원의 'SW교육에 대한 태도'

예비 교원의 SW교육에 대한 정의적 영역 평가를 통해 'SW교육에 대한 태도'를 분석하고자 하였다. 설문 문항은 5단계 리커트 척도(전혀그렇지않다, 그렇지않다, 보통, 그렇다, 매우그렇다)를 사용하여 10개 문항을 만들었다.

<Table 4> Survey Results of Computational Thinking 1

Question	Avg	StDev
Expect to increase computational thinking through software education	3.67	0.937
With the introduction of software education, the need for private tutoring outside of school is expected to increase.	3.49	0.983
Less likely to have difficulty using programming tools.	2.7	1.144
I think that software education should apply different teaching and learning methods.	3.85	0.845
I believe that software education is a fundamental competency that students should have in their future society.	3.99	0.879
I think that students should continue to improve their skills in software.	3.92	0.814
I think I have enough expertise to conduct software education.	2.7	1.03
I think that a variety of teaching and learning methods should be developed for good software education.	3.99	0.888
I believe that a dedicated software teacher is needed to run a software education.	3.98	0.858
In order to educate a small number of students, the placement of an assistant teacher (practical assistant) is necessary.	3.78	0.877

예비 교원의 'SW교육에 대한 태도' 설문에 대한 분석 결과는 <Table 4>와 같다. 예비교원들이 '프로그래밍 도구를 사용하는데 어려움을 느끼는 경우가 적다'는 항목에서 평균 2.7 로 프로그래밍 도구 사용에 어려움을 느끼고 있음을 알 수 있다. 또한 '나는 소프트웨어(SW)교육을 실시할 수 있는 충분한 전문성을 가지고 있다고 생각 한다'

항목에서 평균 2.7 로 소프트웨어 교육을 실시할 수 있는 전문성을 향상시켜야 함을 인지하고 있다고 나타났다.

<Table 5> Survey Results of Computational Thinking 2

SW Class	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
Exist(Avg)	3.68	3.39	2.78	3.66	3.95	3.88	3.02	3.85	3.9	3.76
None(Avg)	3.69	3.51	2.68	3.92	4.01	3.95	2.55	4.04	4.01	3.79

예비교원이 대학교 들어오기 전에 SW수업을 받은 학생들과 대학교를 들어온 후 SW교육을 처음 받은 학생들의 'SW교육에 대한 태도'를 교차 분석한 결과는 <Table 5>와 같다. 대부분의 항목에서는 유의미한 차이가 보이지 않지만, '나는 소프트웨어(SW)교육을 실시할 수 있는 충분한 전문성을 가지고 있다고 생각한다.' 항목에서 SW수업을 들었던 학생 3.02, 대학에서 처음 접하는 학생 2.55 로 많은 차이를 보였으며, 독립표본 T검정을 한 결과 Levene의 등분산 검정 F(5.487), 유의확률(0.021)로 나와 등분산을 가정한 상태에서 평균의 동일성에 대한 T검정 결과 T값(2.497), 유의확률(0.014)로 귀무가설이 기각되어 SW수업을 들었던 학생이 SW수업을 듣지 않았던 학생에 비해 전문성을 가지고 있다고 나왔다.

### 3.4. 예비 교원의 컴퓨팅 사고력

예비교원의 컴퓨팅 사고력을 조사하기 위해 3개의 문제(피보나치 수열, 성적처리하기, 블록 쌓기)를 해결하는 것으로 하였다. 이 문항에 대한 설문결과는 <Table 6>와 같다.

첫 번째 문제는 피보나치 수열에 관련된 문제로 두 개의 문항을 만들었는데, <Table 6>에서 같이 문항1, 문항2에 해당한다. 문항1은 설계 능력의 패턴 찾기, 문항2는 구현 능력의 자동화에 관한 컴퓨팅 사고력을 측정하였다. 자주 접하는 문제로 패턴 찾기는 89.1%, 자동화는 67.4%의 정답률을 보였다. 패턴 찾기는 잘 했으나, 자동화는 실제 결과를 유추하는 것으로 패턴을 실제로 적용하는 데 힘들어 함을 유추할 수 있다.

두 번째 문제는 성적처리하기에 관련된 문제로 세 개의 문항을 만들었는데, <Table 6>에서 같이 문항3, 문항4, 문항5에 해당한다. 문항3은 분석능력의 추상화, 문항4는 분석능력의 자료 분석, 문항5는 설계능력의 알고리즘에 관한

<Table 6> Survey Results of Computational Thinking 3

No	SW Ability	Computational Thinking	Correct		Wrong	
			Frequency	Ratio	Frequency	Ratio
1	Design ability	Find pattern	123	89.1%	15	10.9%
2	Implementation ability	Automation	93	67.4%	45	32.6%
3	Analysis ability	Abstraction	55	39.9%	83	60.1%
4	Analysis ability	Data analysis	4	2.9%	134	97.1%
5	Design ability	Algorithm	86	62.3%	52	37.7%
6	Design ability	Find pattern	26	18.8%	82	59.4%
7	Implementation ability	Automation	89	64.5%	49	35.5%

컴퓨팅 사고력을 측정하였다. 설문 결과 추상화는 39.9%, 자료 분석 2.9%, 알고리즘은 62.3%의 정답률을 보였다. 추상화는 문제가 어떤 것을 처리하고자 하는 것을 찾는 것인데 낮은 정답률을 보였다. 자료 분석은 문제로부터 처리해야 할 자료를 정의할 변수의 수를 묻는 것인데 극소수의 학생만 정답에 반응하였다. 이는 수업 시간에 자료 처리에 관련된 부분이 적은 문제를 많이 다루어서 그와 같은 결과를 낳음을 유추할 수 있다. 알고리즘도 문제 해결 순서를 유추하는 것인데 많은 학생들이 오답을 보였다.

세 번째 문제는 블록 쌓기에 관련된 문제로 두 개의 문항을 만들었는데, <Table 6>에서 같이 문항6, 문항7에 해당한다. 문항6은 설계능력의 패턴 찾기, 문항7은 구현 능력의 자동화에 관한 컴퓨팅 사고력을 측정하였다. 자주 접하는 문제로 패턴 찾기는 18.8%, 자동화는 64.7%의 정답률을 보였다. 패턴 찾기는 문제 해결의 실마리를 찾는 것으로 주어진 문제의 패턴을 만드는데 힘들어 함을 보였다. 자동화는 실제 결과를 유추하는 것으로 패턴을 실제로 찾지 못했지만 실제로 적용하는 것을 유추하였음을 알 수 있었다.

예비교원이 대학교에 들어오기 전에 SW수업을 받은 학생들과 대학교를 들어온 후 SW교육을 처음 받은 학생들의 컴퓨팅 사고력을 비교하기 위해, 문항에서 정답인 학생은 1, 틀린 학생은 2로 처리한 후 이를 교차 분석한 결과는 <Table 7>과 같다. 교차 분석 결과 SW교육을 받은 학생과 대학교에서 SW교육을 처음 받은 학생들 간의 평균 차이는 별로 나지 않았다.

<Table 7> Survey Results of Computational Thinking 4

SW Class	P15	P16	P17	P18	P19	P20	P21
Exist(Avg)	1.05	1.39	1.63	1.98	1.31	1.58	1.39
None(Avg)	1.13	1.3	1.58	1.97	1.38	1.48	1.33

SW수업 유무와 컴퓨팅 사고력의 평균 동일성에 대한 T검정을 수행한 결과 <Table 8>과 같이 두 집단 간의 유의미한 차이가 없는 것으로 나타났다.

<Table 8> Survey Results of Computational Thinking 5

	Levene's equal variance test		T test for equality of means			
	F	Significance	t	Degrees of freedom	Significance Probability (both sides)	
Q1	Assume equal variance	8.252	0.005	-1.348	135	0.18
	Did not assume equal variance			-1.585	112.24	0.116
Q2	Assume equal variance	3.135	0.079	1.002	135	0.318
	Did not assume equal variance			0.975	71.237	0.333
Q3	Assume equal variance	1.421	0.235	0.552	135	0.582
	Did not assume equal variance			0.556	76.781	0.58
Q4	Assume equal variance	0.189	0.664	0.217	135	0.829
	Did not assume equal variance			0.227	84.164	0.821
Q5	Assume equal variance	2.331	0.13	-0.717	117	0.475
	Did not assume equal variance			-0.727	78.315	0.469
Q6	Assume equal variance	1.776	0.185	0.96	133	0.339
	Did not assume equal variance			0.961	73.596	0.34
Q7	Assume equal variance	1.686	0.196	0.715	134	0.476
	Did not assume equal variance			0.702	72.842	0.485

#### 4. 결론

본 논문에서는 예비교원 컴퓨팅 사고력 현황을 파악하기 위해, SW 역량과 컴퓨팅 사고력과의 관계를 정의하였다. 이를 위해 KERIS(2017)의 ‘2017년도 소프트웨어(SW)교육 효과성 측정도구 개발 연구’를 기초로 하여 SW 역량과 컴퓨팅 사고력과의 관계를 수정 제시하였

다. SW역량의 분석 역량에 컴퓨팅 사고력을 ‘추상화’, ‘자료 분석’, ‘분해’ 로, SW 역량의 설계역량에 컴퓨팅 사고를 ‘패턴 찾기’와 ‘알고리즘’으로 정의하였다. 정의된 컴퓨팅 사고력을 토대로 예비교원의 ‘SW교육에 대한 태도’와 ‘컴퓨팅 사고력’측정을 위한 설문 문항을 만들었다. 설문은 한 학기 동안 엔트리(Entry) 수업을 받은 예비교원을 대상으로 ‘SW교육에 대한 태도’와 ‘컴퓨팅 사고력’에 대한 설문을 실시하였다.

설문분석 결과 예비 교원의 SW교육의 정의적 영역을 평가를 통해 ‘SW교육에 대한 태도’에서 ‘나는 소프트웨어(SW)교육을 실시할 수 있는 충분한 전문성을 가지고 있다고 생각한다.’ 항목에서 예비교원이 대학교 들어오기 전에 SW수업을 받은 학생들과 대학교를 들어온 후 SW교육을 처음 받은 학생들 간의 유의미한 차이가 발생하였다. 또한 컴퓨팅 사고력 문제에 있어서는 SW역량의 분석 능력에 해당하는 컴퓨팅 사고력 자료 분석에 있어서 극소수의 학생만 정답을 제출 하였다. 또한 컴퓨팅 사고력의 ‘패턴 찾기’와 ‘추상화’ 분야도 많은 학생들이 오답을 제출하였다.

설문분석 결과를 토대로 예비교원의 컴퓨팅 사고력을 분석해보면 컴퓨팅 사고력 중 추상화, 자료 분석, 자동화에 대한 부분의 개선이 필요함을 보였다. 미래인재인 학생들이 4차 산업혁명 시대를 주도적으로 이끌며 문제를 해결하기 위해서는 컴퓨팅 사고력을 배양해야 한다. 컴퓨팅 사고력 배양의 핵심은 교육 변화의 중심 역할을 하는 예비교원으로서 SW 교육에 대한 인식 개선과 컴퓨팅 사고력에 대한 경험이 풍부할 수 있도록 역량을 성숙시켜야 한다. 이를 위해서는 예비교원의 SW역량과 컴퓨팅 사고력 배양을 할 때 엔트리 등을 활용한 코딩 위주의 교육에서 벗어나야 한다. 예비교원의 컴퓨팅 사고력(추상화, 분해, 패턴인식, 알고리즘, 등)을 배양하기 위해서는 SW를 만드는 과정인 분석과 설계 과정을 정확히 인식하고 이 과정에 적용되는 컴퓨팅 사고력을 습득할 수 있도록 할 필요가 있다. 특히 설문 분석 결과를 기반으로 문제 해결 시 필요한 컴퓨팅 사고력 중 추상화, 자료 분석, 자동화에 대한 부분의 개선에 대한 방법을 보완할 필요가 있다. 이를 통해 체계적인 컴퓨팅 사고력을 갖춘 예비교원을 양성할 수 있는 토대를 구축할 수 있을 것이다.

## References

- [1] Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. In Annual American Research Association meeting, Vancouver, BC, Canada.
- [2] Choi, Hyungshin. (2017). Domestic Literature Review on Computational Thinking Development through Software Programming Education. *JET*, 34(3), 743-774
- [3] Choi, Hyungshin. (2016). Developing Pre-service Teachers' Computational Thinking : Analysis of the Five Core CT Competencies. *Journal of The Korean Association of Information Education*, 20(6), 553-562.
- [4] Denning, P. & Martell, C. (2015). *Great principles of computing*, The MIT Press.
- [5] Jeon, Soojin, Han, Seongkwan. (2016). Descriptive Assessment Tool for Computational Thinking Competencies. *Journal of The Korean Association of Information Education*, 20(3), 255-262.
- [6] Jeong, Inkee. (2017). Study on the Preliminary Teachers' Perception for the Development of Curriculum of the Robot-based Software Education in the Universities of Education. *Journal of The Korean Association of Information Education*, 21(3), 277-284.
- [7] Jin, Seonghee, Rha, Ilju. (2009). A Framework of Teaching Competencies and Comparison of the Perception: between Pre-service and In-service Elementary School Teachers in Korea. *Journal of Elementary Education*, 22(1), 343-368.
- [8] Jo, Miheon. (2018). Analysis of Elementary Pre-service Teachers' Experiences and Understanding of Software Education. *Journal of The Korean Association of Information Education*, 22(1), 81-89.
- [9] Ju Yeojin, Ma, Daisung. (2018). The Development of Abstractable Competency Assessment Standards for the Measurements of Computational Thinking. *Journal of The Korean Association of Information Education*, 22(3), 375-383.
- [10] KERIS. (2017). *A Study on Developing the Evaluating Tool for the Effectiveness of SW Education 2017*.
- [11] Kim, Minjeong, Lee Wongyu, Kim, Jamee. (2017). Presenting the Development Direction Through the Analysis of Tool used to Measure Computational Thinking. *The Journal of Korean Association of Computer Education*, 20(6), 17-25.
- [12] Kim, Seong-sik, Kim, Young-jik, Jo, Ara, Lee, Min-woo. (2019). Development of a Tool for Computational Thinking Assessment in Problem-Solving Programming Education: Paper Type Inspection and Self-Report Questionnaire. *The Journal of Korean Association of Computer Education*, 22(3), 89-99.
- [13] KOFAC. (2015). A Study on Surveying the Actual Conditions and Evaluating the Effectiveness of SW Education in Elementary and Secondary Schools.
- [14] Lee, Jeongmin, Kim, Somang. (2019). Qualitative research of perception and experience of elementary pre-service teachers about SW education. *Journal of The Korean Association of Information Education*, 23(1), 39-53.
- [15] Lee, Hee-Sook, Chung, Jae-Young. (2011). An Analysis of the Influence of Teachers' Traits on Student Achievement - Focusing on Teachers' Efforts to Enhance Professionalism in TIMSS 2007. *The Journal of Korean Teacher Education*, 28(1), 243-266.
- [16] MOE, "Guidelines of Software Education Management", Korea Ministry of Education, 2015.
- [17] National Research Council of the National Academies. (2010). *Report of a workshop on the scope and nature of computational thinking*, The National Academies Press.
- [18] Oh, Jee Hyea, Lee, Soo-Young. (2017). The multi-level modeling analysis on the effects of teachers' teaching competencies on elementary school students' academic achievements, *Journal of Elementary Education*, 28(4), 179-200
- [19] Park, Hyeongyong, Ahn, Seonghun. (2016). A



- Effectiveness Analysis Method for SW Education. *Journal of Creative Information Culture (JCIC)*, 2(2), 83-91.
- [20] Park, Sook-Hyun. (2015). *The effect of practical teaching competence girls' middle school students' perceived on educational achievement : Focused on the mediating effect of attitude towards mathematics*. Theses for Master's Degree, Graduate School of Seoul National University.
- [21] Park, Young-Shin, Hwang, Jin-Kyung. (2017). The preliminary study of developing computational thinking practice analysis tool and its implementation. *Journal of the Korean Society of Earth Science Education*, 10(2), 140-160.
- [22] SRI Education,(2015). Assessment Design Patterns for Computational Thinking Practices in Secondary Computer Science: A First Look.
- [23] Sung, Jungsook, Kim, Hyeoncheol, "Analysis on the International Comparison of Computer Education in Schools. *The Journal of Korean Association of Computer Education*, 18(1), 45-54, 2015.
- [24] Wing M. Jeannette. (2006). Computational Thinking. *"Communications of the ACM"*, 49(3), 33-35.

### 저자소개



#### 김 철

1997 전남대학교 대학원  
전산통계학과(이학박사)  
1998 University of Washington  
(객원교수)  
1992~ 현재 광주교육대학교  
컴퓨터교육과 교수  
관심분야 : 인터넷자원관리, 언플러  
그드활용교육, e-Learning,  
인공지능교육  
e-mail : chkim@gnue.ac.kr