

문제해결 프로그래밍 교육을 위한 범용 컴퓨팅 사고력 척도 개발 연구

이민우[†] · 김성식^{††}

요 약

본 연구의 목적은 교원양성 대학 학부생 대상 문제해결 프로그래밍 교육에서 학습자의 컴퓨팅 사고력을 측정할 수 있는 범용의 검사 도구를 개발하고 타당화 하여 일반화 가능성을 분석하는 것이다. 이를 위해 컴퓨팅 사고력의 평가 영역과 세부 요소를 재정의 하고 이를 평가할 수 있는 자기평가식 범용 컴퓨팅 사고력 검사지를 개발하고, 교원 양성대학의 학생을 대상으로 평가 도구를 적용하여 신뢰도 및 타당도를 분석하였다. 이를 바탕으로 본 연구에서 개발한 자기평가식 범용 컴퓨팅 사고력 검사지의 타당성을 검증하고 일반화 가능성을 확인하였다.

주제어 : 컴퓨팅 사고력, 범용 컴퓨팅사고력 척도, 컴퓨팅사고력 평가, 자기평가식 설문지, 문제해결 프로그래밍 교육

Study on the Development of a General-Purpose Computational Thinking Scale for Programming Education on Problem Solving

Min-Woo Lee[†] · Seong-Sik Kim^{††}

ABSTRACT

The purpose of this study is to develop and validate a general-purpose evaluation tool and to analyze their applicability in problem solving programming education for college students of teacher training college. For this purpose, we have redefined the area of computational thinking and detail elements from the viewpoint of problem solving programming, and developed general-purpose computational thinking scale to evaluate them. The reliability and validity were analyzed by applying the evaluation tool developed for the actual college students of teacher training college. Through this study, it was confirmed that the a general-purpose evaluation tool developed in this study can be used as a tool to computational thinking assessment and can be generalized.

Keywords : Computational Thinking, general-purpose computational thinking scale, Computational Thinking Assessment, Self-Assessment Questionnaire, Problem-Solving Programming Education,

[†]정 회 원: 한국교원대학교 컴퓨터교육과 석사과정
^{††}중심회원: 한국교원대학교 컴퓨터교육과 교수(교신저자)
논문접수: 2019년 9월 10일, 심사완료: 2019년 9월 30일, 게재확정: 2019년 9월 30일

1. 서론

컴퓨팅 교육을 통해 갖추어야 할 핵심 역량으로 문제해결력과 창의력 함양을 위한 컴퓨팅사고력이 제기되고 있다. 교육부에서 발표한 ‘초·중등 SW교육 활성화 방안’에서는 교대 및 사범대학 관련학과의 교육과정엔 소프트웨어 교육을 강화하여 예비교원들이 전문성과 지도역량을 높일 수 있도록 기본 이수과목을 개정하도록 하고 있고[1], 교육부의 ‘2015 개정 교육과정 총론’에 의하면 정보 교과에서는 교과의 주요 역량을 컴퓨팅 사고력, 협력적 문제해결력, 정보문화소양으로 설정하였다. 또한 2015년 교육부에서 발표한 ‘소프트웨어 교육 운영 지침’에 따르면 소프트웨어 교육의 목적으로 컴퓨팅 사고력을 가진 창의, 융합 인재 양성을 제시하고 있다[2][3].

이렇듯 정보 교과 교육에 있어 컴퓨팅 사고력의 중요성은 점점 대두되고 있으며, 컴퓨팅 사고력 함양은 정보 교육의 핵심 목표가 되었다.

더불어 컴퓨팅 사고력은 우리에게 당면한 현실 세계의 복잡한 문제 자체를 이해하고 문제해결 방안을 고안하는 과정이라 볼 수 있기에 컴퓨팅 사고력 발달을 위한 다양한 교수방법, 도구, 이론들은 지금도 활발하게 연구중에 있다.

그렇기에 컴퓨팅 사고력이 얼마나 향상되었는지 교수자와 학습자가 측정 및 확인하여 이를 피드백 자료로 활용할 수 있는 평가 도구가 절실히 필요하지만 컴퓨팅 사고력을 측정하는 것은 그리 간단한 일이 아니다. 컴퓨팅 사고력을 측정하는 것은 단순히 컴퓨터를 잘 사용하거나 소프트웨어를 잘 활용하는 지 평가하는 것이 아닌 프로그래밍을 하기 전 문제를 분석하고, 패턴을 발견하거나 공통된 특성을 추려내어 일반화하는 추상화를 거쳐 알고리즘을 만드는 모든 과정이 포함되어야 하기 때문이다.

더욱이 컴퓨팅 사고력이라는 용어 자체가 비교적 최근에 나왔고 명확한 정의도 약간씩 차이가 있기에 컴퓨팅 사고력 척도 개발 방법이나 사례 역시 표준화된 것이 없는 실정이다.

현재 가장 많이 사용되고 있는 문제지 형태의 컴퓨팅 사고력 검사 도구는 문항이 사전에 공개될 경우 적용의 공정성에 문제가 생길 가능성이 높고,

그러한 우려에 미공개를 전제로 논문이 발표되는 경우가 많아 검사도구의 실효성이 매우 열악한 실정이다. 결국 효과적인 프로그래밍 교육을 위한 범용의 컴퓨팅 사고력 척도 개발이 필요한 상황이다.[4][5][6][7][8]

2015 개정 실과·정보과 교육과정에서는 문제를 해결하기 위해 필요한 프로그래밍을 통해 컴퓨팅 사고력을 신장시키는데 중점을 두고 있다. 더불어 평가에 있어서는 문법 이해와 관련한 지엽적 평가 대신 문제 분석, 추상화, 알고리즘 설계, 프로그램 개발, 수정 등의 수행과정을 종합적으로 평가한다고 제시한다[9].

그리하여 본 연구에서는 컴퓨팅 사고력을 측정할 수 있는 검사 도구, 그 중에서도 SW교육에 있어 큰 부분을 차지하는 프로그래밍 교육의 관점에서 학습자가 문제를 해결하는 과정을 쉽고 빠르게 평가할 수 있는 범용 컴퓨팅 사고력 측정도구를 개발하고자 한다[10].

지금까지 개발된 대부분의 컴퓨팅 사고력 측정도구는 2003년 공개된 PISA의 문제지를 개량하여 연구목적에 맞는 문제지 형태로 개발한 경우가 대부분이다. 이와 같이 컴퓨팅 사고력을 평가하기 위해 검사 도구로서 평가 대상이나 목적을 중심으로 문제지를 개발하여 평가할 수 있지만, 내용이 공개될 경우에 미리 공부해서 응시할 가능성이 있기 때문에 1회성에 불과하다. 따라서 Likert 척도 방식의 설문지를 통해 컴퓨팅 사고력을 측정할 수 있다면 학습자들의 측정 횟수에 관계없이 사용이 가능한 범용 컴퓨팅 사고력 검사 도구가 될 수 있다.

따라서 본 연구에서는 문제지 형태의 측정도구와 동일한 평가 요소를 측정할 수 있는 Likert 척도 방식의 설문지 형태의 측정도구를 동시에 개발하여 2개의 검사도구 사이의 상관관계를 분석함으로써, 궁극적으로 프로그래밍 교육 관점에서 학생들의 컴퓨팅 사고력을 측정할 수 있는 범용의 척도를 개발하는 것을 목적으로 하였다.

개발한 척도는 전국의 7개 교원양성 대학에 적용 후 신뢰도 및 타당도 분석을 통해 척도를 정교화 함과 동시에 일반화 가능성을 연구하며, 적용 결과에 따라 학습자에게 척도 활용법을 제시하고자 한다.

2. 이론적 배경

2.1 컴퓨팅 사고력의 구성요소

Wing(2006)은 컴퓨팅 사고력을 추상화와 자동화로 구분하였으며[11], David Moursund는 컴퓨팅 사고력이 개발, 표현, 시험, 오류 수정의 순서로 구성된 절차적 사고임을 강조하였다[12].

Barr와 Stephenson(2011)은 자료수집, 자료분석, 자료제시, 문제분해, 추상화, 제어구조, 알고리즘 및 절차, 분석과 모델 타당화, 자동화, 병렬화, 시험 및 검증, 시뮬레이션의 12가지 단계로 세분화하여 제시하였고[13], Blake는 컴퓨팅 사고력에 표현, 시각화, 모델링을 포함한다고 주장하였다.[12]

CSTA(2011)는 이러한 의견들을 바탕으로 컴퓨팅 사고력이 6가지 특징을 지니는 하나의 문제해결 과정이라는 견해와 함께 문제해결을 위한 컴퓨팅 사고력의 구성요소를 제시하였다[14]. CSTA의 견해는 우리나라의 소프트웨어교육 운영지침에 반영되었고, 교육부가 제시한 컴퓨팅 사고력의 구성요소는 자료수집 및 분석, 구조화, 추상화, 자동화, 일반화로 세분화 된다[3].

이와 같은 연구들에 기반을 두고 볼 때, 컴퓨팅 사고력은 논리화되고 절차화된 사고를 통하여 컴퓨터의 원리, 개념을 바탕으로 문제들을 해결하는 인지, 정의적 사고 과정이라 할 수 있으며[15], 구성요소는 큰 맥락에서 대동소이하나 세분화 정도에 있어 차이가 있는 것으로 사려된다.

컴퓨팅 사고력을 지나치게 세분화하면 실제 활동시, 교과목표 달성에 어려움을 겪을 수 있기에 적절한 구분과 조작적 정의가 필요하다. 또한 실제 학습 활용에는 하나의 요소만을 가지고 진행 할 수 있고, 때에 따라 여러 요소가 융·복합되어 이루어질 수도 있으므로 구성요소 들이 교육과정에 제시된 순서에 따라 나열되거나 모든 하위요소들을 교육과정에 포함시켜야 할 필요는 없다[16].

따라서 본 연구에서는 문제해결 프로그래밍 교육의 관점에서 컴퓨팅 사고력의 구성요소를 문제분해, 추상화, 알고리즘 절차, 자동화로 제시하고자 한다.

2.2 CT 평가도구 관련 선행연구 분석

컴퓨팅 사고력의 중요성이 강조됨에 따라 컴퓨팅 사고력이 무엇인지에 대한 논의가 계속 되어 왔으나, 이것을 어떻게 측정할 것인가에 대한 연구는 상대적으로 부족하다.

Brenna와 Resnick은 컴퓨팅 사고력의 평가 방법으로 프로젝트 포트폴리오 분석, 산출물 기반 심층 인터뷰, 디자인 시나리오 등을 제시하였고[17], Seiter & Foreman은 산출물을 통해 컴퓨팅 사고력을 증거변수, 디자인 패턴 변수, 컴퓨팅 사고 개념의 3개 측면으로 평가하는 방법을 제안하였다[18]. 하지만 이들의 평가방법은 스크래치 교육 후 기능 및 내용에 대한 평가로써 스크래치를 교육 받지 않은 학생은 사용할 수 없다[17][18].

이은경(2009)의 연구에서 개발한 컴퓨팅 사고력 측정 도구는 컴퓨팅 사고력을 세 영역으로 나누고 각 영역의 하위 요인으로 구성된 검사 도구를 제작하였지만 컴퓨팅 사고력의 영역과 하위요인 선정에 있어 명확한 근거가 나타나지 않았다[5].

안성훈 외(2016)의 연구에서 개발한 컴퓨팅 사고력 측정 도구는 SW교육에서 제시하는 내용에 대한 내용 타당도를 확보하였으며, 초, 중학생용으로 나누어 제작하였다[7]. 하지만 단편적이고 단순한 사고력을 측정하며, 문항 난이도가 대체적으로 쉬운 편이라는 한계가 있다. 양재명 외(2017)의 연구에서 개발한 컴퓨팅 사고력 측정 도구는 기존 안성훈 외(2016)의 연구를 보완하며 고등학생용 검사 도구를 포함하여 제작하였다[8]. 하지만 적용의 공정성을 위해 문제지를 공개하지 않아 활용에 어려움이 있다.

위에서 살펴본 것처럼 컴퓨팅 사고력을 측정하는 도구를 개발하기 위한 연구들이 존재하지만 몇 가지 한계점이 존재하여 학습자들의 컴퓨팅 사고력을 측정하기에는 제한이 있다.

또한 프로그래밍 교육의 관점에서 컴퓨팅 사고력을 평가하는 연구가 부족한 시점에서 본 연구에서는 효과적인 프로그래밍 교육을 위해 측정 횟수나 시기에 관계없이 범용으로 측정이 가능한 컴퓨팅 사고력 검사도구를 개발하고자 하였다.

3. 연구 절차 및 방법

3.1 연구 절차

컴퓨팅 사고력 측정 문항 개발은 프로그래밍 교육의 관점에서 컴퓨팅 사고력을 재정의하는 것으로부터 시작한다. 컴퓨팅 사고력의 조작적 정의를 바탕으로 측정 내용 요소 선정을 위하여 전문가집단을 대상으로 전문가 협의회를 실시한다. 이 과정에서 측정 영역 및 세부요소를 추출하며, 이를 바탕으로 비 구조화된 문제 풀이 방식의 검사 도구를 개발, 적용한다. 이를 통해 수집된 학습자들의 검사 결과를 정선하고, 문헌조사와 연구를 통해 문항 내용요소를 선정한다. 선정한 문항 내용 요소를 바탕으로 Likert 척도 방식의 검사지를 제작한다. 개발한 컴퓨팅 사고력 측정 문항의 적절성을 알아보기 위하여 교원양성대학의 컴퓨터교육과 학부생을 대상으로 본 검사를 실시한다. 검사 결과를 통계적 분석, 신뢰도와 타당도 검증을 통해 문항을 정제하고, 전문가협의를 통해 문항의 내용을 수정·분석한다[19]. 이를 통해 개발된 자기평가식 범용 컴퓨팅 사고력 검사지를 전국 7개 교원양성대학의 컴퓨터교육과 학생에게 적용함으로써, 실험결과의 분석을 통해 타당성과 일반화 가능성을 연구하며, 적용 결과에 따라 학습자의 척도 활용법을 제시한다.

3.2 연구 대상

본 연구는 교원양성대학 컴퓨터교육과 학생을 대상으로 하는 문제해결 프로그래밍 교육에 적용할 컴퓨팅 사고력 평가 도구를 개발하고 일반화 가능성을 검증하기 위한 것이다.

이를 위해 자기평가식 범용 컴퓨팅 사고력 측정 도구의 활용 가능성을 검토하기 위한 1차 연구는 K교원양성대학 학부생 중 문제해결 프로그래밍 관련 강의를 수강중인 컴퓨터 교육과 학생 42명으로 구성하였다.

측정도구의 타당성 및 일반화 가능성을 검토하기 위한 2차 연구에는 전국 7개 교원양성대학의 컴퓨터교육과 학생 105명이 참여하였다.

3.3 연구 방법

| 구분 | 문제해결 프로그래밍교육 관점 컴퓨팅 사고력 검사 도구 | |
|-----|----------------------------------|------------------------|
| | 서술형 평가 | 자기 평가 |
| 형태 | 지필형 검사지 | Likert 5점 척도 방식 설문지 |
| 문항수 | 5개 문항 | 24개 문항 |

↓

K 교원양성대학 컴퓨터교육과 학생 42명

- 지필형 검사지 및 자기평가식 설문지 검사 실시
- 신뢰도 검증 및 상관 분석을 통한 자기평가식 범용 컴퓨팅 사고력 척도의 활용 가능성 분석

↓

전국 7개 교원양성대학 컴퓨터교육과 학생 105명

- 자기평가식 설문지 검사 실시
- 내적일관성 검증 및 요인분석을 통한 자기평가식 범용 컴퓨팅 사고력 척도의 일반화 가능성 및 활용 방법 제시

[그림 1] 컴퓨팅 사고력 평가도구 개발 및 적용

4. 컴퓨팅 사고력 측정 검사도구 개발

4.1 평가 영역 및 핵심요소 구성

ISTE와 CSTA는 컴퓨팅 사고력을 자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화의 9개 세부요소로 나누어 제시하였다[20]. 하지만 문제해결 프로그래밍 교육의 관점에서 컴퓨팅 사고력을 평가할 때 위에서 제시한 구성요소를 모두 포함하기에는 제약이 있다. 한국과학창의재단(2014)은 컴퓨팅 사고력의 세부 구성요소들이 반드시 배열된 순서에 따라 제시되거나, 모든 하위요소를 교육과정에 포함시킬 필요가 없다고 하였다[16]. 이에 프로그래밍 교육에 필요한 평가 영역 및 핵심요소를 재구성하여 연구를 진행하였다. 2015 개정교육과정 정보교과의 문제해결과 프로그래밍 영역 핵심 개념을 바탕으로 본 연구에서는 문제 해결 프로그래밍교육의 평가 영역으로 문제분해, 추상화, 알고리즘 절차, 자동화를 선정하고, 이에 대한 평가도구 개발에 집중하였다. 평가 영역별 세부요소는 각 3개로 나누어 총 12개의 세부요소로 구성하였으며, 그 조작적 정의는 <표 1>과 같다.

<표 1> 프로그래밍 교육의 관점에서
컴퓨팅 사고력 핵심 요소

| 영역 | 세부요소 | 조작적 정의 내용 |
|---------|---------|----------------------------------|
| 문제분해 | 문제이해 | 문제가 요구하는 입·출력 상황을 이해하는 능력 |
| | 문제분석 | 문제 난이도와 해결 방안을 파악하는 능력 |
| | 문제분할 | 문제를 해결 가능한 작은 문제로 분할하는 능력 |
| 추상화 | 패턴인식 | 문제에서 반복적 규칙, 공통적 속성등을 탐색하여 찾는 능력 |
| | 핵심요소 추출 | 불필요한 요소를 제거하고 핵심요소를 추출하는 능력 |
| | 모델링 | 추출된 핵심요소들을 글이나 그림으로 표현하는 능력 |
| 알고리즘 절차 | 알고리즘 설계 | 문제해결에 필요한 연산을 구조화하는 능력 |
| | 알고리즘 표현 | 필요한 연산들을 의사코드 또는 자연어로 표현하는 능력 |
| | 알고리즘 분석 | 필요한 연산들을 수행시간 관점에서 분석하는 능력 |
| 자동화 | 프로그래밍 | 컴퓨터가 이해 가능한 언어로 구현할 수 있는 능력 |
| | 자료구조 | 문제를 해결하는데 필요한 자료구조를 이해 및 활용하는 능력 |
| | 디버깅 | 문제해결 과정 및 결과를 확인 후 오류를 수정하는 능력 |

4.2 지필형 검사지의 개발

4.2.1 예비 문항 개발

<표 1>에서 선정한 세부요소와 조작적 정의를 기준으로 컴퓨팅 사고력을 측정하기 위한 지필형 검사지를 개발한다. 검사지의 문항들은 프로그래밍 언어를 이해하였는지 여부를 확인하는 것 외에도 주어진 문제들을 해결하기 위해 적합한 방법을 탐색하고 컴퓨팅 사고력 발달 정도를 평가할 수 있도록 개발한다. 문제는 4대 영역, 12개 요소를 모두 측정할 수 있도록 평가지표를 제작하여 개발하였으며, 한 문항에 3~4개의 소문항으로 이루어진 실기 및 서술형 문제로 구성하였다.

문항은 기본적으로 문제해결 프로그래밍 수업에

서 활용가능한, 자료구조와 알고리즘의 기본 개념이 포함된 내용들로 구성하였다. <표 2>는 지필형 검사지 예비 문항의 구성이다.

<표 2> 지필형 검사지 예비문항 구성

| 제목 | 컴퓨팅 개념 | 소문항 수 |
|-------|-----------------|-------|
| 우편배달부 | 그래프, 탐색 | 3 |
| 무술대회 | 정렬 | 3 |
| 하노이 탑 | 동적계획법, 재귀 | 5 |
| 우박수 | 동적계획법, 프로그래밍 언어 | 3 |
| 다트게임 | 배열, 프로그래밍 언어 | 4 |

4.2.2 타당도 검증 및 최종 평가 도구 개발

개발한 문항의 내용 타당도를 검증하기 위해 전문가 협의회를 5차에 걸쳐 실시하였다. 전문가 집단은 현직교사로서 관련 연구로 학위를 취득하였거나, 컴퓨팅 사고력 관련 연구를 진행중인 박사과정 3인, 석사과정 3인과 1명의 전문가로 구성되었다.

예비문항의 문항난이도, 컴퓨팅 개념의 적절성, 평가요소의 적절성, 문항 명확성 등에 대하여 논의 후 수정, 보완 및 소문항 수의 조절을 거쳐 총 5문항의 지필형 검사 방식의 평가도구를 개발하였다. 개발한 지필형 검사지의 평가지표는 다음과 같다 (<표 3> 참고).

<표 3> 지필형 검사지의 평가 지표

| 문제 | 척도 | 문제분해 | | | 추상화 | | 알고리즘 절차 | | | 자동화 | | | |
|-------|----|------|------|------|------|--------|---------|--------|--------|--------|-------|------|-----|
| | | 문제이해 | 문제분석 | 문제분할 | 패턴인식 | 핵심요소추출 | 모델링 | 알고리즘설계 | 알고리즘표현 | 알고리즘분석 | 프로그래밍 | 자료구조 | 디버깅 |
| 우편배달부 | 1 | ○ | ○ | | | | | | | | | | |
| | 2 | | | | | | ○ | ○ | | | | | |
| | 3 | | | | | | | | | | ○ | ○ | |
| 무술대회 | 1 | ○ | ○ | | ○ | | | | | | | | |
| | 2 | | | | | | ○ | ○ | | | | ○ | |
| | 3 | | ○ | | | | ○ | | | | | | |
| 하노이 탑 | 1 | ○ | ○ | | | ○ | | | | | | | |
| | 2 | | | ○ | ○ | | ○ | | | | | | |
| | 3 | | | | | | ○ | ○ | | | | | |
| | 4 | | | | | | | ○ | ○ | | ○ | ○ | |

| 문제 | 척도 | 문제분해 | | | 추상화 | | 알고리즘 절차 | | | 자동화 | | | |
|-------|----|-------|------|------|------|------|---------|-----|--------|--------|--------|-------|------|
| | | 세부 요소 | 문제이해 | 문제분석 | 문제분할 | 패턴인식 | 핵심요소추출 | 모델링 | 알고리즘설계 | 알고리즘표현 | 알고리즘분석 | 프로그램밍 | 자료구조 |
| 우박수 | 1 | ○ | ○ | | | | | | | | | | |
| | 2 | | | | | ○ | | ○ | | | | ○ | |
| | 3 | | | | | | | | | ○ | ○ | | ○ |
| 다트 게임 | 1 | ○ | ○ | | | | | | | | | | |
| | 2 | | ○ | | | | ○ | | | | | | |
| | 3 | | | | | | | ○ | ○ | ○ | ○ | ○ | ○ |
| | 4 | | | | ○ | ○ | | | | | ○ | | ○ |
| 요소별합계 | | 5 | 7 | 1 | 3 | 3 | 5 | 5 | 3 | 3 | 4 | 3 | 5 |
| 척도별합계 | | 13 | | | 11 | | 11 | | | 12 | | | |

4.3 자기평가식 설문지의 개발

4.3.1 예비 문항 개발

<표 1>을 바탕으로 제작된 지필형 문제지 형태의 측정도구와 동일한 평가 요소를 측정할 수 있는 Likert 5단계 평정 척도의 자기평가식 설문지를 개발하였다. 자기평가식 설문지는 측정 횟수에 관계 없이 범용으로 적용이 가능하도록 특정 프로그래밍 언어나 환경에 제약을 받지 않는 문항들로 구성하였다. 개발한 예비 문항은 문제분해 영역 6문항, 추상화 영역 6문항, 알고리즘 절차 영역 6문항, 자동화 영역 6문항의 총 24문항으로 이루어져 있다.

4.3.2 타당도 검증 및 최종 평가 도구 개발

3차에 걸친 전문가 협의회를 통해 문항의 내용 타당도를 검증한 뒤 문항을 구체화하고, 표현 수정 및 불필요한 내용들을 제거해 가는 과정을 진행하였다. 또한 각각의 문항들이 해당 요소의 조작적 정의들을 모두 포함하는지에 유의하여 문항을 수정, 보완하였다.

교원 양성 대학의 문제해결 프로그래밍 교육을 위한 자기평가식 컴퓨팅 사고력 측정도구에 최종 선정된 문항은 문제분해 영역 6문항, 추상화 영역 6문항, 알고리즘 절차 영역 6문항, 자동화 영역 6문항의 총 24문항으로 <표 4>와 같다.

<표 4> 최종 선정 문항

| 영역 | 최종 문항 |
|---------|---|
| 문제이해 | <ol style="list-style-type: none"> 1. 나는 주어진 문제 상황에서 문제의 현재 상태를 파악할 수 있다. 2. 나는 주어진 문제 상황에서 도달해야 할 목표 상태를 파악할 수 있다.. 3. 나는 문제가 주어진 조건 하에 해결가능한 문제인지 알 수 있다. 4. 나는 문제를 어떻게 해결해야 할지 대략적인 방법을 구상할 수 있다. 5. 나는 복잡한 문제를 해결 가능한 작은 문제들로 나누어 생각할 수 있다. 6. 나는 복잡한 문제를 해결 가능한 수준의 작은 문제로 나누어 해결해 본 경험이 있다. |
| 추상화 | <ol style="list-style-type: none"> 1. 나는 주어진 문제 상황에서 반복적인 규칙들을 탐색하여 찾아낼 수 있다. 2. 나는 주어진 문제 상황에서 공통적인 속성들을 탐색하여 찾아낼 수 있다. 3. 나는 문제 해결에 필요한 요소와 불필요한 요소를 구분 지을 수 있다. 4. 나는 문제에서 불필요한 부분을 없애고 필요한 내용만 변수로 선언할 수 있다. 5. 나는 문제상황에서 추출된 핵심요소를 이해하기 쉬운 간단한 글이나 그림으로 표현할 수 있다. 6. 나는 문제상황에서 추출된 핵심요소들을 간단하게 도식화 할 수 있다. |
| 알고리즘 절차 | <ol style="list-style-type: none"> 1. 나는 문제해결에 필요한 연산들을 조건,반복,제어 구조를 사용하여 설계할 수 있다. 2. 나는 문제해결에 필요한 연산들을 단계적으로 나타낼 수 있다. 3. 나는 문제를 해결하기 위한 알고리즘을 자연어나 의사코드로 표현할 수 있다. 4. 나는 문제를 해결하기 위한 알고리즘을 순서도로 표현할 수 있다. 5. 나는 문제를 해결하기 위한 연산의 수행시간이 어느 정도인지 짐작할 수 있다. 6. 나는 주어진 문제에 대하여 다양한 해결방법을 찾아내 수행 시간으로 비교할 수 있다. |
| 자동화 | <ol style="list-style-type: none"> 1. 나는 하나 이상의 프로그래밍 언어를 기술적으로 잘 다룰 수 있다. 2. 나는 자신이 사용하는 프로그래밍 언어를 활용하여 문제를 해결할 수 있다. 3. 나는 스택, 트리, 그래프 등 자료구조 별 특징을 이해하고 있다. 4. 나는 주어진 문제를 해결할 때 어떤 자료구조가 적합한지 판단하고 활용할 수 있다. 5. 나는 입·출력 데이터를 통해 프로그램의 정확성을 점검할 수 있다.. 6. 나는 프로그램이 문제를 해결하지 못할 때, 수정 및 보완을 통해 이를 해결할 수 있다. |

5. 연구 결과

5.1 평가도구의 신뢰도 검증 및 상관분석

연구의 대상은 문제해결 프로그래밍 관련 수업을 수강중인 교원양성대학 컴퓨터교육과 학부생 42명으로 하였다. 개발한 검사지가 평가하려는 것을 일관성있게 평가 할 수 있는지 신뢰도 분석을 진행한 후, 문제지와 검사지의 유사성을 검증하고, 검사지가 범용으로 적용 가능한지에 대한 검증을 위해 문제지와 검사지에 대한 상관분석을 실시하였다.

5.1.1 신뢰도 검증

신뢰도는 피실험자들에게 동일한 검사를 반복 실시하였을 때 평가의 일치성이다. 즉, 평가하려는 것을 얼마나 안정적으로 일관성있게 평가하였는지를 나타낸다[21]. 일반적으로 Cronbach α 계수가 .700 이상일 경우 '신뢰도가 양호하다'라고 이야기 할 수 있다. 지필형 검사지와 자기평가식 설문지의 신뢰도는 <표 5>와 같다.

<표 5> 지필형 검사지의 신뢰도

| 구분 | Cronbach's α |
|-----------|---------------------|
| 지필형 검사지 | .714 |
| 자기평가식 설문지 | .974 |

분석 결과 지필형 검사지와 자기평가식 설문지의 Cronbach's α 는 .714와 .974로 평가 도구로 사용을 하기에 신뢰도가 모두 적절한 것으로 나타났다.

자기평가식 설문지의 경우 문항을 제거할 경우 α 계수가 높아지는 문항이 존재하지 않았다. 이를 통해 자기평가식 설문지의 문항들이 평가하고자 하는 컴퓨팅사고력을 일관적이고 정확하게 검사하고 있다고 할 수 있다. 따라서 제거할 문항 없이 문항 수를 24문항으로 유지하였다.

5.1.2 상관 분석

일반적으로 동일한 요소를 서로 다른 도구를 이

용해 평가할 경우, 평가 도구 사이의 상관관계를 통해 결과가 얼마나 일치하는지 검증하는데, 이러한 과정을 수렴 타당도 검증이라고 한다. 이때 두 결과 사이의 상관계수 값이 높으면 두 검사는 동일한 특성들을 측정함을 의미한다.

자기평가식 설문지의 타당도를 제대로 확보하기 위해서는 두 검사도구를 동시에 투입하여도 그 순서에 상관없이 타당도가 확보되어야 한다. 따라서 지필형 검사지와 자기평가식 설문지의 상관관계를 명확히 분석하기 위해 두 그룹으로 나누어 검사를 실시한다.

그룹 A는 지필형 검사지를 먼저 실시한 후 자기평가식 설문지를 실시하고, 나머지 그룹 B는 자기평가식 설문지 실시 후 지필형 검사지를 시행한다. <표 6>은 상관분석 결과이다.

<표 6> 상관분석 결과

| 구분 | 상관계수 | 유의확률 |
|--------------------------------|--------|------|
| A 그룹 (지필형 검사지 -> 자기보고식 설문지) | .669** | .001 |
| B 그룹 (자기평가식 설문지 -> 지필형 검사지) | .667** | .001 |
| 전체 | .673** | .000 |

상관분석 결과 A, B 두 그룹의 지필평가와 자기평가식 설문지의 점수는 유의수준 .01에서 유의한 상관이 있으며, 상관계수는 .673으로 상관이 높다.

두 그룹의 상관계수가 각각 .669, .677로 모두 상관이 높은 것으로 보아 지필형 검사지, 자기평가식 설문지의 실시 순서에 상관없이 컴퓨팅 사고력이 일정하게 평가된다는 것을 알 수 있다.

이는 본 연구를 통해 개발한 지필형 검사지와 자기평가식 설문지가 문제해결 프로그래밍 교육의 관점에서 컴퓨팅 사고력의 동일한 특성을 평가하고 있음을 의미하며, 문제해결 프로그래밍 능력도 평가에 반영하고 있음을 알 수 있다.

5.2 자기평가식 컴퓨팅 사고력 척도 요인분석

상관분석을 통하여 도출된 결과를 바탕으로 지필형 검사지와 자기평가식 설문지가 문제해결 프로그

래밍 교육의 관점에서 컴퓨팅 사고력의 동일한 특성을 평가함을 알 수 있다.

이를 통해 Likert 척도 형식으로 제작된 설문지 형태의 검사도구가 사용자의 사용횟수나 실시 순서에 구애 받지 않고 적용 가능한 범용 컴퓨팅 사고력 척도로 활용 가능함을 검토할 수 있다.

자기평가식 검사지의 일반화 가능성 및 범용 컴퓨팅 사고력 측정 도구로서의 타당도를 파악하고자 전국 7개 교원 양성대학 컴퓨터교육과 학생 105명을 대상으로 신뢰도분석과 요인분석을 실시하였다.

5.2.1 평가 영역별 신뢰도 분석

컴퓨팅사고력의 평가영역으로 선정한 문제분해, 추상화, 알고리즘 절차, 자동화 항목의 내적 일관성 검증을 위해 신뢰도 분석을 실시한다.

주로 크론바흐 알파 계수를 산출하여 신뢰성을 판단하는데, 일반적으로 0.700 이상이면 신뢰도가 양호한 것으로 판단한다. 측정도구의 평가영역별 신뢰도는 <표 7>과 같다.

<표 7> 평가 영역별 신뢰도 분석

| 변수 | Cronbach's alpha | 항목 개수 |
|--------|------------------|-------|
| 문제분해 | .927 | 6 |
| 추상화 | .921 | 6 |
| 알고리즘절차 | .944 | 6 |
| 자동화 | .908 | 6 |

문제분해, 추상화, 알고리즘 절차, 자동화 항목에 대하여 각각 크론바흐 알파 계수를 산출한 결과, 모두 변수가 .700 이상으로 높게 나타나 주요 변수들의 신뢰도는 양호한 것으로 판단되었다. 따라서 검사의 신뢰도를 방해하는 문항은 없는 것으로 평가되어, 문항 제거 없이 분석을 진행하였다.

5.2.2 요인 분석

요인 분석(Factor analysis)은 여러개의 측정 항목을 공통 요인으로 묶어 자료의 복잡함을 줄이고 변수를 구성하는 항목들이 동일한 구성 개념을 측

정하고 있는지를 파악하는 분석방법이다. 문제해결 프로그래밍 교육을 위한 자기평가식 컴퓨팅사고력 측정도구의 타당도를 파악하고자 요인분석을 실시하였다.

요인 추출 방법으로 주축요인 추출을 실시하였고, 베리맥스 회전을 하였다. 그 결과 KMO 측도는 .853으로 나타났고, Bartlett의 구형성 검정 결과 역시 유의확률 .05 미만으로 나타나 요인분석 모형이 적합한 것으로 판단되었다. 한편 누적분산이 70.910%로 나타나, 구성된 4개의 요인은 설명력이 높은 것으로 판단되었다.

각 요인에 구성된 항목들을 보면 각각의 요인에 6개의 항목이 포함되어 있다. 구성된 항목들의 내용을 바탕으로, 첫 번째 요인은 자동화, 두 번째 요인은 문제분해, 세 번째 요인은 추상화, 네 번째 요인은 알고리즘 절차로 명명하였다. 요인 적재 값이 모두 0.4 이상으로 나타났기에, 전반적인 측정도구의 타당도를 만족하였으며, 추가적인 항목의 제외 및 조정 없이 분석을 진행하였다. 측정도구의 요인분석 결과는 <표 8>과 같다.

<표 8> 측정도구의 타당도(요인분석) 분석

| 요인 | 1 | 2 | 3 | 4 |
|-------|------|------|------|------|
| 자동화2 | .759 | .227 | .061 | .330 |
| 자동화5 | .740 | .132 | .041 | .250 |
| 자동화6 | .699 | .227 | .083 | .196 |
| 자동화1 | .694 | .260 | .136 | .272 |
| 자동화4 | .679 | .311 | .193 | .158 |
| 자동화3 | .636 | .227 | .266 | .299 |
| 문제분해5 | .190 | .895 | .201 | .233 |
| 문제분해6 | .185 | .884 | .193 | .242 |
| 문제분해4 | .182 | .880 | .210 | .240 |
| 문제분해1 | .302 | .657 | .257 | .157 |
| 문제분해2 | .219 | .588 | .243 | .071 |
| 문제분해3 | .298 | .567 | .118 | .088 |
| 추상화3 | .119 | .215 | .948 | .134 |
| 추상화2 | .115 | .214 | .939 | .115 |
| 추상화1 | .094 | .239 | .936 | .120 |
| 추상화4 | .241 | .187 | .634 | .253 |
| 추상화6 | .102 | .165 | .527 | .492 |
| 추상화5 | .151 | .247 | .444 | .340 |

| 요인 | 1 | 2 | 3 | 4 |
|--|--------|--------|--------|--------|
| 알고리즘절차3 | .363 | .221 | .187 | .815 |
| 알고리즘절차5 | .375 | .226 | .183 | .807 |
| 알고리즘절차4 | .401 | .190 | .178 | .647 |
| 알고리즘절차1 | .548 | .168 | .227 | .587 |
| 알고리즘절차6 | .510 | .240 | .247 | .584 |
| 알고리즘절차2 | .491 | .238 | .240 | .570 |
| 아이젠값 | 4.655 | 4.342 | 4.202 | 3.819 |
| 공통분산(%) | 19.396 | 18.092 | 17.510 | 15.911 |
| 누적분산(%) | 19.396 | 37.488 | 54.999 | 70.910 |
| KMO=.853 , Bartlett's $\chi^2=3587.809$ (p<.001) | | | | |

6. 결론 및 제언

6.1 결론

본 연구는 문제 해결 프로그래밍 교육을 받고 있는 예비교원들의 컴퓨팅 사고력을 평가하기 위한 범용 평가 도구 개발을 목표로 하였다.

이 연구를 통해 첫째, 문제해결 프로그래밍의 관점에서 컴퓨팅 사고력을 문제분해, 추상화, 알고리즘 절차, 자동화의 4가지 영역으로 나누어 조작적으로 정의하고, 각각의 세부요소를 선정하였다.

둘째, 이를 바탕으로 문제해결 프로그래밍 관점의 지필형 문제를 제작하여 컴퓨팅 사고력의 세부영역을 측정하였다. 전문가 협의회를 통해 문항 난이도, 컴퓨팅 개념 적절성, 평가요소 적절성, 문항 명확성 등의 논의를 거쳐 최종적으로 5문항의 컴퓨팅 사고력 측정 지필형 검사지를 제작하였다.

셋째, 수행된 문제지 형태의 측정도구와 동일한 평가요소를 측정할 수 있는 Likert 척도 방식의 자기평가식 측정도구를 개발하여 2개의 검사도구 사이의 상관관계를 분석함으로써 궁극적으로 프로그래밍 교육 관점에서 학생들의 컴퓨팅 사고력을 측정할 수 있는 범용의 척도를 개발하고자 하였다.

상관분석을 통해 타당도를 검증한 결과 지필형 검사지와 자기평가식 설문지 사이에 상관 관계가 높은 것으로 나타났다. 더욱이 지필형 검사지와 자기평가식 설문지의 투입 순서에 상관없이 상관계수가 .6 이상으로 높게 측정되었다. 이는 likert 척도 형식으로 제작된 자기평가식 측정도구가 사용자의

사용횟수나 실시 순서에 영향을 받지 않고 적용 가능한 범용 컴퓨팅 사고력 척도로 활용가능함을 시사한다.

넷째, 개발된 척도를 7개 교원 양성대학의 컴퓨터전공 학부생 105명에게 적용하여 척도의 타당성 검증 및 일반화 가능성을 검토하였다. 평가영역으로 선정한 네 개 영역의 신뢰도 분석 결과 모든 영역에서 .700 이상의 Cronbach α 계수가 산출되어 주요 변수들의 신뢰도는 모두 양호한 것으로 판단되었다.

자기평가식 컴퓨팅사고력 측정도구의 변수를 공통적 요인으로 묶어 자료의 복합성을 줄이고, 측정 변수들이 동일 구성 개념을 측정하는지 파악하기 위하여 요인분석을 실시한 결과 통계적으로 적합한 모형으로 판단되었고, 구성된 4개 요인의 설명력 역시 높은 것으로 측정되었다. 요인 적재값 역시 모두 .400 이상으로 나타나 측정 도구로서의 타당도를 만족하였다. 이는 자기평가식 컴퓨팅사고력 측정도구가 타당성이 있는 척도임을 시사한다.

지필형 검사지와 상관분석과 전국 7개 교원양성대학 학생들을 대상으로 실시한 검사의 요인분석 결과 본 연구에서 개발한 자기평가식 범용 컴퓨팅 사고력 척도는 문제 해결 프로그래밍을 학습하고 있는 예비 교원들의 컴퓨팅 사고력을 범용으로 평가하는데 많은 도움을 줄 수 있을 것이다.

6.2 활용계획

자기평가식 검사도구만으로 컴퓨팅사고력 측정을 완전히 대체하기에는 많은 문제점이 있는 것이 사실이다. 설문자가 자기 자신에 대하여 응답하는 경우 반응을 의도적으로 바꿀 수 있는 '반응적 자기보고 변화 요인'이 발생할 수도 있고[22], 개인의 자신감과 같은 인성적 측면에 따라 같은 컴퓨팅 사고력 수준에도 상이한 응답이 나올 수 있다.

몇몇 연구에서 보고된 바와 같이 자기평가와 다른 형태의 평가가 항상 긍정적인 상관관계를 갖는 것도 아니며, 대부분의 자기평가지는 성공적으로 수행을 했든 아니든 간에, 특정 점수대에 몰리는 경향, 즉 중간 분포의 경향이 있다고 한다.

그러나 컴퓨팅 사고력은 그 특성상 인지적 작용

의 과정이기 때문에 단순히 실험자나 설문자의 행동을 관찰하거나 문제지 형식의 검사 도구를 통해 측정하기에는 한계가 있다. 이런 경우 표면적 행동 측정을 보완해 줄 수 있는 방법의 하나가 바로 자기평가이다.

자기평가식 컴퓨팅 사고력 측정 도구를 활용하여 설문자의 컴퓨팅사고력에 대한 메타인지적 분석이 가능하다고 판단되며, 설문자는 자기반성을 통한 학습 동기의 유지나 유발이 가능하다고 생각된다.

또한 개인의 학습도중 점진적인 변화 정도 측정이 가능하다. 본 연구를 통해 개발한 자기평가식 컴퓨팅 사고력 측정도구는 사용자의 사용 횟수나 실시 순서에 구애받지 않고 범용으로 적용 가능하기에 학습이 진행되는 도중 수 회 측정하여 개인이 컴퓨팅 사고력 구성요소의 어떤 능력이 향상되었는지 확인이 가능하다.

마지막으로 설문지로서의 척도 활용 가능성 이외에 컴퓨팅 사고력 관련 교수학습 과정이나 평가 과정에서도 활용이 가능하다. 제시된 척도들은 컴퓨팅 사고력의 중요한 구성 요소로 컴퓨팅 사고력 관련 교수학습 상황이나 문제, 학습 과정을 설계할 때 참고할 수 있는 좋은 지침으로 활용할 수 있다. 또한 활동지와 같은 교재를 개발하거나 평가할 때에도 척도를 기준으로 고려할 수 있을 것이다. 추가적인 컴퓨팅 사고력 평가 도구의 개발과정에서 본 연구에서 제작한 척도를 참조할 수 있다.

6.3 제언

본 연구를 통하여 문제해결 프로그래밍 교육을 위한 자기보고식 컴퓨팅 사고력 측정 도구 및 지필형 검사지를 개발하였다. 이를 바탕으로 향후 지속적인 개선 및 연구를 위하여 다음과 같이 제언하고자 한다.

첫째, 본 연구에서 개발한 평가 도구는 교원양성 대학의 예비교원을 대상으로 개발하였다. 자기 자신을 스스로 평가하는 경우 지능이 높거나, 성취가 높을수록 평가 결과가 정확하다[23]. 따라서 인지능력이 아직 성숙치 못한 초, 중등 학생을 대상으로 일반화 할 때는 주의하여야 한다.

더불어 컴퓨터를 전공하지 않은 비전공자 및 초,

중,고 학생들을 대상으로 하는 컴퓨터 소양교육에서의 컴퓨팅 사고력 평가도구를 제작시 문항 내 컴퓨터 용어를 줄이고 일상생활의 언어를 사용하되 컴퓨팅 사고력의 구성요소를 충분히 이끌어 낼 수 있는 설문지를 만들어야 할 것이다.

둘째, 본 연구는 문제해결 프로그래밍 교육의 관점에서 컴퓨팅 사고력 측정도구의 평가영역을 설정하고, 세부 요소를 조작적으로 정의하였다. 즉, 이 연구의 결과물은 문제해결 프로그래밍 교육에 대한 컴퓨팅 사고력 평가에 한정된 것이며, 다른 영역으로 확대하기에는 한계가 있다.

셋째, 평가 참여자가 같은 학교급에 있더라도 수준이나 지식의 정도차이가 심할 수 있다. 측정도구를 학교급별로 제작하는 것도 좋은 생각이지만 설문지의 수준을 다양하게 만들어 컴퓨팅 사고력에 등급을 부여하는 것도 새로운 대안이 될 수 있다.

넷째, 컴퓨팅 사고력의 구성요소는 다면성을 가지고 있기에 단일 방법의 평가만으로 컴퓨팅 사고력 수준의 완벽한 측정 및 평가는 어렵다는 점을 인정하고 다양한 방법의 평가들이 상호보완적으로 활용되고, 이들 간의 수렴타당도를 보고하는 연구들의 공유가 계속되어야 하겠다.

본 연구가 새로운 컴퓨팅 사고력 측정도구의 개발 필요성 및 기존 컴퓨팅 사고력 평가 도구의 개선 방향에 대한 시사점을 제시하는데 도움이 되길 기대하는 바이다.

참 고 문 헌

- [1] 교육부 (2016). 초·중등 SW교육 활성화 방안. 교육부 보도자료
- [2] 교육부 (2015). 2015 개정 교육과정 총론. 교육부 고시 제2015-74호 (별책 1)
- [3] 교육부 (2015). 소프트웨어 교육 운영지침. 교육부 고시
- [4] 최정원 · 이영준 (2014). 컴퓨팅 사고력 평가 방안 설계. 한국컴퓨터정보학회 논문지, 22(2), 177-178.
- [5] 이은경 (2009). Computational Thinking 능력 향상을 위한 로봇 프로그래밍 교수 학습

모형. 박사학위논문. 한국교원대

[6] 김병수 (2015). **계산적 사고력 신장을 위한 PPS 기반 프로그래밍 교육 프로그램**. 박사학위논문. 제주대학교

[7] 안성훈 외 (2016). **2016년도 SW교육 연구학교 효과성 분석 연구**. 한국교육학술정보원. 연구보고 KR2016-4

[8] 양재명 외 (2017). **2017년도 소프트웨어(SW) 교육 효과성 측정도구 개발 연구**. 한국교육학술정보원. 연구보고 CR2017-8

[9] 교육부 (2015). **실과(기술·가정)/정보과 교육과정**. 교육부 고시 제2015-74호 (별책 10)

[10] 최정원 · 이은경 · 김경훈 · 이영준 (2015). 2015 개정 정보 교과 교육과정에서 학습자의 컴퓨팅 사고력 평가 방안에 대한 제언. **한국컴퓨터교육학회 논문지**. 19(2), 9-12.

[11] Wing, J. M. (2006). Computational Thinking. *Communication of the ACM*, 19(3), 33-35.

[12] NATIONAL_RESEARCH_COUNCIL (2010). Report of a Workshop on the Scope and Nature of Computational Thinking. The National Academies Press.

[13] Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community *ACM Inroads*, 2(1), 48-54.

[14] Computer Science Teachers Association (2011). *CSTA K-12 Computer Science Standards*.

[15] 최무영 (2015). **초등학생의 컴퓨팅 사고력 능력 신장을 위한 찾아가는 융합인재교육 프로그램 개발**. 석사학위논문. 경인교육대

[16] 이영준 · 백성혜 · 신재홍 · 유현창 · 정인기 · 안상진 · 최정원 · 전성균 (2014). **초중등 단계 Computational Thinking 도입을 위한 기초 연구**. 한국과학창의재단.

[17] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking.

[18] Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of

computational thinking of primary grade students. *In Proceedings of the ninth annual international ACM conference on International computing education research*, pp.59-66. ACM.

[19] 엄명용 · 조성우 (2005). **사회복지실천과 척도 개발**. 서울: 학지사.

[20] ICTE & CSTA(2011). *CSTA K-12 Computer Science Standards Revised*.

[21] 성태제 · 시기자 (2006). **연구방법론**. 서울: 학지사.

[22] 유진은 (2015). **양적연구 방법과 통계분석**. 서울: 학지사.

[23] Mabe, P. A., & West, S. G. (1982). Validity of self-evaluation of ability: A review and meta-analysis, *Journal of applied psychology*, 67(3), 280-296.



이 민 우

2007 청주교육대학교
사회과교육과(교육학학사)
2018~현재 한국교원대학교
컴퓨터교육과 석사과정
관심분야: 소프트웨어 교육, 메이커 교육
E-Mail: rhddlr01@gmail.com



김 성 식

1994 고려대학교
경영학과(경영학사)
1977~1991 교육부 및
교육정책자문위원회 근무
1988 미국오리곤주립대학교 대학원
컴퓨터과학과(이학석사)
1992 고려대학교 컴퓨터학과(이학박사)
1992~현재 한국교원대학교 컴퓨터교육과 교수
관심분야: 교육용 콘텐츠, 알고리즘, 원격교육
E-Mail: seongkim@knue.ac.kr