

컴퓨팅 사고력에 관한 탐색을 통한 소프트웨어 교육의 지식, 기능, 태도적 측면에 관한 고찰

전용주†

† 청주 사직초등학교

A Study on the Cognitive, Psychomotor and Affective Aspects of Software Education by Exploring Computational Thinking

Yong-Ju Jeon†

† Sagik Elementary School

요 약

2015 개정 교육과정 총론에서는 Bloom의 교육목표분류에 기반하여 지식, 기능, 태도 영역에서 균형 있는 수업과 평가를 하도록 강조하고 있으며, 국외에서도 컴퓨팅 교육에서 지식, 기능, 태도 등에서 균형있는 접근을 하기 위한 노력을 기울이고 있는바, 현재 제안되고 있는 대부분의 수업 및 평가가 지식 측면을 중심으로 구성되어 있는 점을 보완하기 위하여 본 연구에서는 컴퓨팅 사고력에 대한 국외의 동향과 2015 개정 교육과정 총론에 비추어 소프트웨어 교육의 지식, 기능, 태도적 측면의 의미와 다룰 수 있는 내용을 고찰하였다.

1. 서 론

소프트웨어 교육 운영지침과 2015 개정 교육과정의 고시 이후 소프트웨어 교육에 대한 논의와 연구가 활발하게 진행되고 있다. 특히, 소프트웨어 교육 운영지침과 개정 교육과정에서는 소프트웨어 교육을 ‘컴퓨터 과학의 기본적인 개념과 원리를 기반으로 다양한 문제를 창의적이고 효율적으로 해결하는 컴퓨팅 사고력(Computational Thinking, CT)을 기르는 교육’으로 정의하여[1], 현장에서는 이를 수업, 평가 등에서 구현하기 위한 다양한 노력을 기울이고 있다.

그러나, 현재 논의되고 있는 컴퓨팅 사고력에 기반한 소프트웨어 교육의 수업과 평가 사례를 살펴보면, 지식적인 측면의 내용을 강조하는 경우가 대부분인 것을 알 수 있다. 즉, 컴퓨팅 사고력과 관련된 개념, 지식 등을 익히기 위한 수업, 이를 확인하기 위한 평가가 주를 이루고 있다[2][3][4].

2015 개정 교육과정 총론에서는 Bloom의 교육목표분류에 기반하여 지식, 기능, 태도 측면에서 균형 있는 수업과 평가를 하도록 강조하고 있으며[5], 국외에서도 컴퓨팅 교육에서 지식, 기능, 태도 등에서 균형있는 접근을 하기 위한 노력을 기울이고 있는바[6][7][8], 본 연구에서는 컴퓨팅 사고력에 대한 국외의 동향과 2015 개정 교육과정 총론에 비추어 소프트웨어 교육의 지식, 기능, 태도적 측면의 의미와 다룰 수 있는 내용을 고찰하여 후속적인 평가 연구 수행에 대한 기초 자료를 마련하고자 하였다.

2. 이론적 배경

2.1 컴퓨팅 사고력의 조작적 정의[6]

컴퓨팅 사고력에 관한 연구는 Seymour Papert가 1980년대에 그의 저서에서 철차적 사고라는 용어를 사용한 이후, 2006년 Wing의 연구에 의해 컴퓨팅 사고력(Computational Thinking, CT)이라는 용어가 재조명되면서 컴퓨터 교육 현장에서 이슈화되기 시작하였다[9]. 이후 수많은 학자와 다양한 단체에서 컴퓨팅 사고력의 정의를 내리고 그 속성을 규명하기 위한 연구를 진행하였다.

이후 컴퓨팅 사고력의 다양한 정의를 유·초·중등 교육에 효과적으로 적용할 수 있는 기반을 마련하고자, 국제교육공학협회(ISTE)와 미국컴퓨터교사협회(CSTA)에서 컴퓨터 과학 교사, 연구자 및 관련 참여자들의 검토 결과를 종합하여, 컴퓨팅 사고력에 대한 조작적 정의와 교육적 적용 방안을 명확히 하였다. 이들이 제시한 컴퓨팅 사고력의 조작적 정의는 <표 1>과 같다[6].

<표 1> 컴퓨팅 사고력의 조작적 정의[6]

구분	조작적 정의
컴퓨팅 사고력이 갖는 특징	- 문제해결을 위해 컴퓨터나 도구를 사용할 수 있도록 문제를 표현하기 - 논리적으로 자료를 조직하고 분석하기 - 모델이나 시뮬레이션 등의 추상화를 통해 자료를 표현하기

	<ul style="list-style-type: none"> - 알고리즘 사고와 같은 일련의 절차를 통해 해결책을 자동화하기 - 가장 효율적이고 효과적인 절차와 자원을 조합하여 목표를 달성하는 데 필요한 해결책을 확인하고, 분석하고, 구현하기 - 이러한 문제 해결 절차를 다양한 문제에 일반화하고 적용하기
컴퓨팅 사고력 태도성향, 태도적 차원	<ul style="list-style-type: none"> - 복잡한 문제를 다루는 데 있어서의 자신감 - 어려운 문제를 다루는 인내력 - 모호성에 대한 허용 - 답이 정해지지 않은 개방형 문제를 다룰 수 있는 능력 - 공동의 목표나 해결책을 달성하기 위해 다른 사람과 의사소통하고 일할 수 있는 능력



<그림 1> CAS의 컴퓨팅 사고력[8]

2.2 MIT 스크래치 팀(2012)의 컴퓨팅 사고력[7]

Brennan & Resnick (2012)은 당시 MIT Media Lab 소속 연구자로서 스크래치를 활용한 상호작용적 산출물을 제작하는 과정에서 어린나이의 학습자들을 대상으로 인터뷰 및 관찰을 통한 연구를 진행하였고, 이를 토대로 개념(Concepts), 실행(Practices), 관점(Perspectives)의 측면에서 컴퓨팅 사고력을 제정의하였다(<표 2> 참고). 이는 교육과정에서 교육목표를 분류하는 지식, 기능, 태도 관점과 연결할 수 있다는 점에서 매우 유용한 접근 관점이라고 할 수 있다.

<표 2> 컴퓨팅 사고력의 핵심 영역과 구성요소[7]

핵심 영역	구성요소
computational concept (개념)	sequences, loops, events, parallelism, conditionals, operators, data
computational practice (실행)	being incremental and iterative, testing and debugging, reusing and remixing, abstracting and modularizing
computational perspectives (관점)	expressing, connecting, questioning

2.3 영국 CAS의 컴퓨팅 사고력[8]

영국의 CAS에서는 컴퓨팅 사고력이 단순히 컴퓨터처럼 생각하는 것을 의미하는 것이 아니라, [그림 1]과 같은 개념과 접근을 포함하고 있다고 설명하고 있다[8].

이를 구체적으로 살펴보면 컴퓨팅 사고력 개념으로 컴퓨터 과학의 기초를 이루고 있는 개념인 논리(logic), 알고리즘(algorithms), 분해(decomposition), 패턴(patterns), 추상화(abstraction), 평가(evaluation)등을 제시하고 있다. 또한 컴퓨팅 사고력에 대한 접근으로 시도하기(tinkering), 창작하기(creating), 디버깅(debugging), 계속 도전하기(persevering), 협업하기(collaborating)를 제시하고 있다.

CAS가 제시한 컴퓨팅 사고력 또한 지식, 기능, 태도적 측면에 관계된 내용들을 찾아볼 수 있다.

3. 소프트웨어 교육의 지식, 기능, 태도적 측면 고찰

본 연구는 컴퓨팅 사고력에 관한 탐색을 통한 소프트웨어 교육의 지식, 기능, 태도적 측면을 고찰하기 위한 연구로, 앞에서 살펴본 이론적 배경에 기반하여 각 측면의 의미와 내용을 고찰해보고자 한다.

교육과정에서 지식 영역은 인지적 영역으로 개념에 대한 이해, 적용, 분석 등을 다루게 되며, 기능 영역은 심동적 영역으로 실행 및 구현에 관한 내용을 다루게 된다. 또한 태도 영역은 정의적 영역으로 주의, 반응, 가치화, 조직화, 인격화 등에 관련된 내용을 다루게 된다[10]. 따라서, 이를 토대로 소프트웨어 교육의 지식, 기능, 태도 측면의 의미를 고찰해 보면 <표 3> 과 같다.

<표 3> 소프트웨어 교육의 지식, 기능, 태도 측면의 의미 고찰

소프트웨어 교육의 지식, 기능, 태도 측면	의미
지식 측면	소프트웨어 교육을 통해 이해해야 할 개념 및 이를 적용하고 분석하는 내용까지 포함함
기능 측면	소프트웨어 교육을 통해 실행하고 구현할 수 있는 내용을 포함함
태도 측면	소프트웨어 교육을 통해 함양할 수 있는 태도, 가치, 반응, 내면화 등의 내용을 포함함

또한 이론적 배경에서 살펴본 국외 사례의 하위 내용을 소프트웨어 교육에서 다룰 수 있는 지식, 기능, 태도적 측면의 내용으로 유목화하여 고찰한 결과는 <표 4>과 같다.

<표 4> 소프트웨어 교육의 지식, 기능, 태도 측면에서 다를 수 있는 내용 고찰

소프트웨어 교육의 지식, 기능, 태도 측면	컴퓨팅 사고력의 조작적 정의 (ISTE & CSTA, 2011)	MIT 스크래치 팀의 컴퓨팅 사고력 (Brennan & Resnick, 2012)	CAS의 컴퓨팅 사고력 (CAS, 2014)
지식 측면	<ul style="list-style-type: none"> - 문제해결을 위해 컴퓨터나 도구를 사용할 수 있도록 문제를 표현하기 - 논리적으로 자료를 조직하고 분석하기 - 모델이나 시뮬레이션 등의 추상화를 통해 자료를 표현하기 	<개념> <ul style="list-style-type: none"> - sequences(순차) - loops(반복) - conditionals(조건) - events(이벤트) - parallelism(병렬처리) - operators(연산자) - data(자료) 	<개념> <ul style="list-style-type: none"> - 논리(logic) - 알고리즘(algorithms) - 분해(decomposition) - 패턴(patterns) - 추상화(abstraction) - 평가(evaluation)
기능 측면	<ul style="list-style-type: none"> - 알고리즘 사고와 같은 일련의 절차를 통해 해결책을 자동화하기 - 가장 효율적이고 효과적인 절차와 자원을 조합하여 목표를 달성하는 데 필요한 해결책을 확인하고, 분석하고, 구현하기 - 이러한 문제 해결 절차를 다양한 문제로 일반화하고 적용하기 	<실행> <ul style="list-style-type: none"> - being incremental and iterative (점진적이고 반복적인 시도) - testing and debugging (테스팅과 디버깅) - reusing and remixing (재사용과 재조합) - abstracting and modularizing (추상화와 모듈화) 	<접근> <ul style="list-style-type: none"> - 시도하기(tinkering) - 창작하기(creating) - 디버깅하기(debugging) - 계속 도전하기(persevering) - 협업하기(collaborating)
태도 측면	<ul style="list-style-type: none"> - 복잡한 문제를 다루는 데 있어서의 자신감 - 어려운 문제를 다루는 인내력 - 모호성에 대한 허용 - 답이 정해지지 않은 개방형 문제를 다룰 수 있는 능력 - 공동의 목표나 해결책을 달성하기 위해서 다른 사람과 의사소통하고 일할 수 있는 능력 	<관점> <ul style="list-style-type: none"> - expressing(표현하기) - connecting(연결하기) - questioning(질문하기) 	

4. 결론 및 제언

본 연구는 컴퓨팅 사고력에 대한 국외의 동향과 2015 개정 교육과정 총론에 비추어 소프트웨어 교육의 지식, 기능, 태도적 측면의 의미와 다를 수 있는 내용을 고찰하기 위한 연구로, 이를 통해 후속적인 평가 연구 수행에 대한 기초 자료를 마련하고자 하였다.

이와 관련된 후속 연구로, 본 연구자는 소프트웨어 교육의 지식, 기능 태도적 측면의 영역별 및 각 영역이 통합된 수업 구현 및 평가 구현에 관한 연구를 수행할 예정이다.

본 연구가 소프트웨어 교육의 지식, 기능, 태도적 측면에 관련된 후속 연구의 유용한 기초자료로 활용되길 바란다.

참고 문헌

[1] 교육부 (2015). **소프트웨어 교육 운영 지침**.
 [2] 에듀넷 (2016). **2015 연구학교 운영 결과 보고서**.
http://www.edunet.net/redu/swedusvc/softwareEduStdyForm.do?class_id=CLSS0000036270&menu_id=0181
 [3] 한국교육학술정보원 (2016). **컴퓨팅 사고력 측정도구**.

[4] 한국교육학술정보원 (2017). **컴퓨팅 사고력 알아보기(초등학교)**.
 [5] 교육부 (2015). **2015 개정 교육과정 총론 해설**.
 [6] ISTE & CSTA (2011). *Computational Thinking Leadership Toolkit 1st edition*.
<http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>
 [7] Brennan, K. & Resnick, M. (2012). *New framework for studying and assessing the Development of Computational Thinking, Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada*.
 [8] Computing At School. (2014). *What is computational Thinking?*
<https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/>
 [9] Wing J. M. (2006). *Computational thinking*. *Communication of the ACM*, 49(3), 33-35.
 [10] 박성익, 임철일, 이재경, 최정임 (2011). *교육 방법의 교육공학적 이해*. 파주: 교육과학사.