초·중등 정보교육과정과 Computational Thinking 평가요소에 관한 연구

권 정 인* 김 재 현**

◈ 목 차 ◈

1. 서 론

미래사회의 핵심역량을 갖춘 인재교육에 있어서 소프트웨어 중심의 의사소통능력, 문제해결력, 정보 활용 기술 등은 필수 핵심 능력이다[9]. 이러한 인재교육의 변화에 발맞춰 최근 우리나라를 비롯한 미국, 일본, 이스라엘, 인도, 영국 등에서는 소프트웨어를 기반으로 문제해결력을 향상시키고 정보과학의 원리를 익힐 수 있는 체계적인 교육내용들을 제시하고 있다[9]. 이와 같이 소프트웨어 교육의 열풍에 보다 박차를 가하고 있는 개념으로 Wing(2006)에 의해 일반화된 Computational Thinking에 대한 연구가 소프트웨어 교육중심의 문제해결력 향상에 중요한 요소로 부각되고 있다. Computational Thinking이란 문제해결방법에 대한 사고의 과정으로 모든 사람이 읽기, 쓰기, 계산하기를 배우고 익히며 학습하는 것과 마찬가지로 Computational Thinking도 배우고 학습해야 한다고 주장하면서 소프트웨어 교육의 새로운 방향을 제시하고 있다[6]. 따라서 미래사회의 핵심역량을 갖춘 우수한 소프트웨어 분야의 기술 및 인재를 발굴하고 지속적으로 육성하기 위해서는 Computational Thinking기반의 문제해결에 대한 교육과 교육과정이 서로 연계되어야 하며 그 평가 역시 융합적인 요소를 포함해야 한다[7]. 그러나 과거 우리의 교육과 교육과정에서는 소프트웨어 교육보다는 소프트웨어 활용교육에 더 치중하였으며, 문제해결력 보다는 기술 숙지 교육의 교육과정과 평가가 주를 이루었다. 또한, 사회제도의 대부분이 기본적인 ICT 기술 숙지 능력에 대한 검증을 요구할 뿐 Computational Thinking과 문제해결력에 대한 요구는 없었다. 그러나, 최근 미래사회의 핵심 역량인 창의력과 문제해결 능력을 겸비한 창의적 인재 양성을 목표와 초·중등학생들의 ICT핵심역량을 객관적으로 평가하고자 소프트웨어 교육 운영 지침을 교육부가 발표한 후 초·중등학교에서 소프트웨어 역량에 대한 교수학습방법 및 평가 운영방안에 관한 관심이 집중되었다[7].

본 연구에서는 초·중등학생들의 미래핵심역량 중 하나인 소프트웨어 역량의 함양을 위해 교육부가 발표한 소프트웨어 교육 운영 지침의 성취기준을 바탕으로 Computational Thinking 주요개념의 적용 사례에 대해 제안을 하고자 한다. 제안된 최종모델은 미래사회의 핵심역량을 갖춘 소프트웨어 인재 양성의 이정표를 제시하고, 학교현장의 소프트웨어 교육의 활성화를 이를 킬 수 있다.

2. 관련 연구

2.1. Computational Thinking의 개요

Computational Thinking은 Seymour Papert(1996)가 기하학적 아이디어 창출을 위해 사용한 방법으로 우리에게 소개되었다[5]. Wing(2006)은 Computational Thinking

* 성균관대학교 소프트웨어학과
** 성균관대학교 컴퓨터교육과
☆ 본 연구는 미래창조과학부 및 정보통기술진흥센터의 SW 중심대학지원사업의 연구결과로 수행되었음(R7115161004000l002)

은 수학적 사고와 결합하면 문제를 해결할 수 있고, 공학적 사고와 결합하여 복잡한 문제를 분해할 수 있으며, 과학적 사고와 결합하여 우리가 이해하고 계산이 가능하도록 접근 할 수 있게 해주는 분석적 도구이다라고 하였다[6]. 최근 Barr, Harrison, & Coney(2011)의 연구 결과를 토대로 CSTA와 ISTE(2011)에서는 Computational Thinking의 주요개념을 (표 1)과 같이 제시하였다[1].

(표 1) Computational Thinking의 주요개념[1]

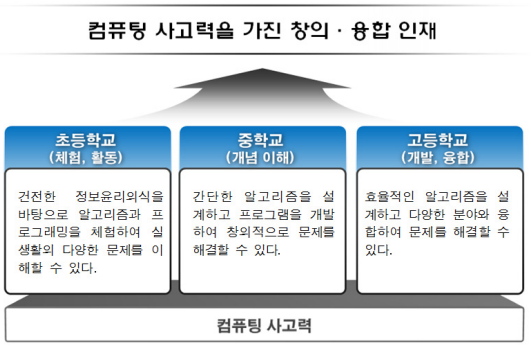
개념	정의
자료수집	문제의 이해와 분석을 토대로 문제를 해결하기 위한 자료를 모으는 단계
자료분석	수집된 자료와 문제에 주어진 자료를 밭캉으로 자료를 분류하고 분석하는 단계
자료표현	문제의 자료 내용을 그래프, 차트, 단어, 이미지 등으로 표현하는 단계
문제 분해	문제를 해결해나가기 위해 문제를 나누어 분석하는 단계
추상화	문제의 복잡도를 줄이기 위해 기본 주요 개념의 정의를 설정하는 단계
알고리즘과 절차	지금까지 문제를 해결하기 위한 과정을 절차적 단계로 표현하는 단계
자동화	컴퓨팅 기기가 수행할 수 있도록 해결과정을 알고리즘화하는 단계
시뮬레이션	문제해결을 위한 실험모델을 만드는 단계
병렬화	문제 해결의 공동에 목표를 달성하기 위한 작업을 수행하는 단계

Computational Thinking은 첫째, 문제 해결을 위해 사고력과 컴퓨팅을 융합하여 문제를 해결할 수 있도록 절차화한다. 둘째, 자료를 논리적으로 구조화하고 분석한다. 셋째, 모델링과 시뮬레이션과 같은 추상화를 통해 자료를 표현한다. 넷째, 알고리즘적인 사고를 통해 해결과정을 자동화한다. 다섯째, 자료의 처리과정 중 가장 효과적이고 효율적인 과정을 선택해 해결책을 만들고 구현한다. 여섯째, 문제해결 과정의 다양한 문제 상황을 반복적으로 적용하여 어렵고 복잡한 문제를 일반적인 문제로 일반화한다. 와 같은 특징을 가지는 하나의 문제 해결 과정이라고 보고 있다[4]. 이와 유사하게 Bundy(2007)는 Computational Thinking 개념은 문제 해결 과정을 통해 다른 학문분야의 교육에서도 사용되어 왔으며, 계산적 사고력은 모든 학문분야에서 기본적으로 필요하다고 주장하였다[2]. Cuny, Snyder & Wing (2010)은 최근 Computational Thinking에 대한 정의를 문제의 정보처리 과정을 효과적으로 수행될 수 있는 형태로 표현될 수 있도록 문제와 그 해결책을 재구성하는 것에 관련된 하나의 사고 과정이 라고 기술하고 있다[3]. 즉, Computational Thinking이란 문제 해결에 대한 단순한 컴퓨팅의 활용이 아닌 창의적이고 융합적인 차원에서 사고하는 것을 의미한다. 또한, 해결과정을 컴퓨팅 시스템으로 처리될 수 있는 형태로 구조화하고, 알고리즘화하는 것으로 논리적 사고의 과정이라고 할 수 있다. 뿐만 아니라, 문제 해결을 위한 컴퓨팅의 기본 개념을 이용하여 컴퓨터를 활용할 수 있는 인간의 인지 능력을 포함하는 기술이라고 볼 수 있으며, 이러한 기술은 정보사회의 모든 분야에서 기본적으로 필요한 능력이다.

2.2. 소프트웨어 교육 운영 지침

교육부에서는 소프트웨어 교육 운영지침을 발표하고 초·중등학교에서 이루어지는 소프트웨어 교육은 프로그램 개발 역량보다는 실생활의 문제를 Computational Thinking로 해결할 수 있도록 하는 것에 역점을 둔다고 하였다[7]. 즉 앞으로의 소프트웨어교육은 개념과 원리 중심의 교육이 아닌 실생활 문제해결 중심의 교육으로 정보사회의 필수적 요소인 Computational Thinking의 필요성과 중요성을 인지하고 그 가치를 경험할 수 있게 교육방법을 개선하겠다는 뜻이다.

소프트웨어 교육에서 추구하는 인재상은 학습자로 하여금 도전적이고 복잡한 문제를 해결하는 능력에 절



(그림 1) 소프트웨어 교육이 추구하는 인재상[7]

(표 2) 소프트웨어 교육 내용요소

영역	초등학교	중학교	고등학교
생활과 소프트웨어	• 나와 소프트웨어 - 소프트웨어와 생활 변화	• 소프트웨어의 활용과 중요성 - 소프트웨어의 종류와 특징 - 소프트웨어의 활용과 중요성	• 컴퓨팅과 정보 생활 - 컴퓨팅 기술과 융합 - 소프트웨어의 미래
	• 정보 윤리 - 사이버공간에서의 예절 - 인터넷중독과 예방 - 개인정보 보호 - 저작권 보호	• 정보 윤리 - 개인정보 보호와 정보 보안 - 지적 재산의 보호와 정보 공유	• 정보 윤리 - 정보 윤리와 지적 재산 - 정보 보안과 대응 기술
		• 정보기기의 구성과 정보 교류 - 컴퓨터의 구성 - 네트워크와 정보 교류	• 정보기기의 동작과 정보 처리 - 정보기기의 동작 원리 - 정보처리의 과정
알고리즘과 프로그래밍	• 문제 해결 과정의 체험 - 문제의 이해와 구조화 - 문제 해결 방법 탐색	• 정보의 유형과 구조화 - 정보의 유형 - 정보의 구조화	• 정보의 표현과 관리 - 정보의 표현 - 정보의 관리
		• 컴퓨팅 사고의 이해 - 문제 해결 절차의 이해 - 문제 분석과 구조화 - 문제 해결 전략의 탐색	• 컴퓨팅 사고의 실제 - 문제의 구조화 - 문제의 추상화 - 모델링과 시뮬레이션
	• 알고리즘의 체험 - 알고리즘의 개념 - 알고리즘의 체험	• 알고리즘의 이해 - 알고리즘의 이해 - 알고리즘의 실제	• 알고리즘의 실제 - 복합적 구조의 알고리즘 실제 - 알고리즘의 분석과 평가
	• 프로그래밍의 체험 - 프로그래밍의 이해 - 프로그래밍의 체험	• 프로그래밍의 이해 - 프로그래밍 언어의 이해 - 프로그래밍의 기초	• 프로그래밍의 이해 - 프로그래밍 언어의 분류
			• 문제해결과 프로그래밍 - 프로그래밍의 실제
컴퓨팅과 문제해결		• 컴퓨팅 사고 기반의 문제 해결 - 실생활의 문제 해결 - 다양한 영역의 문제 해결	• 컴퓨팅 사고 기반의 융합 활동 - 프로그래밍과 융합 - 팀프로젝트의 제작과 평가

차적인 체계를 갖출 수 있도록 도울 수 있는 교육이다[8]. 또한, 확산적사고와 발산적사고 능력을 확장시키기 위한 과정과 절차를 통해 창의적인 문제해결 방법들을 점차적으로 발전시킬 수 있는 기회를 갖을 수 있는 교육이다[8]. 이에 우리의 교육부에서는 '소프트웨어 교육 운영 지침'을 통해 학교급별 목표와 내용체계를 (표 2)와 같이 제시하였다[7].

3. Computational Thinking반영 교수-학습 방법 제안

3.1 생활과 소프트웨어 영역

일상생활의 소프트웨어 사례를 통해 정보사회의 기술과 앞으로 우리생활의 변화에 대해 파악한다. 소프트웨어와 함께 관련된 다양한 직업세계를 살펴보고 자신의 적성에 맞는 소프트웨어의 진로를 찾아 볼 수 있게 교수-학습을 구성한다. 정보윤리를 통해 개인 정보보호와 지적재산권의 보호와 공유에 대해 토론할 수 있는 기회를 갖고 실생활에 적용할 수 있는 학습을 할 수 있도록 유도한다[7]. Computational Thinking의 주요개념 중 자료수집과 자료 분석을 위주로 실생활의 소프트웨어 적용사례에 대해 학습할 수 있는 교수-학습을 구성한다.

3.2 알고리즘과 프로그래밍 영역

생활과 소프트웨어 영역을 통해 우리생활 속 소프트웨어의 필요성을 인지했다면 실제 소프트웨어 중 정보의 처리 과정과 유형을 일상생활의 사례를 통해 탐색해 보고, 다양한 정보의 구조화 방법을 학습한다. 정보의

(표 3) 소프트웨어 교육 내용요소와 Computational Thinking의 반영

소프트웨어 교육 운영 지침 및 평가요소 반영			
영역	중영역	내용 요소	평가요소
생활과 소프트웨어	소프트웨어의 활용과 중요성	소프트웨어의 종류와 특징	자료수집 자료분석
		소프트웨어의 활용과 중요성	자료수집 추상화
	정보윤리	개인정보 보호와 정보보안	자료수집 자료분석
		지적 재산의 보호와 공유	문제분석 병렬화
	정보기기의 구성과 정보 교류	컴퓨터의 구성	자료수집 추상화
		네트워크와 정보 교류*	문제분석 추상화
알고리즘과 프로그래밍	정보의 유형과 구조화	정보의 유형	자료수집 자료분석
		정보의 구조화*	자료분석 자료표현
	컴퓨팅 사고의 이해	문제 해결 절차의 이해	자료표현 문제분해
		문제 분석과 구조화	문제분해 추상화
		문제 해결 전략의 탐색	문제분해 알고리즘과 절차
	알고리즘의 이해	알고리즘의 이해	문제분해 알고리즘과 절차
		알고리즘의 설계	알고리즘과 절차 병렬화
	프로그래밍의 이해	프로그래밍 언어의 이해	알고리즘과 절차 자동화
		프로그래밍의 기초	알고리즘과 절차 자동화
컴퓨팅과 문제해결	컴퓨팅 사고 기반의 문제해결	실생활 문제 해결	문제분해 시뮬레이션
		다양한 영역의 문제 해결	시뮬레이션 병렬화

구조화를 통해 실제 문제 해결의 구조화 방법을 적용할 수 있게 학습하고, 이를 프로그래밍 언어로 구현하는 방법에 대해 교수-학습을 구성한다. 알고리즘과 프로그래밍 영역에서는 정보의 유형별 입력방법과 문제해결의 절차적 나열, 절차적 나열의 오류 검토를 위해 프로그래밍 언어를 사용하는데 학습의 방향을 둔다[7]. Computational Thinking의 주요개념 중 문제분해, 추상화, 알고리즘과 절차의 요소를 위주로 교수-학습을 구성한다.

3.3 컴퓨팅과 문제해결 영역

생활과 소프트웨어 영역을 통해 소프트웨어의 필요성을 인지하고, 알고리즘과 프로그래밍 영역을 통해 소프트웨어의 처리 방법을 학습했다면 컴퓨팅과 문제해결의 영역을 통해 그간 학습한 내용을 바탕으로 실생활의 여러 가지 문제 중 컴퓨팅을 적용해 해결할 수 있는 사례를 직접 해결 할 수 있도록 교수-학습을 구성한다[7]. Computational Thinking의 주요개념 중 실생활의 문제분해와 시뮬레이션, 병렬화의 요소를 중심으로 교수-학습을 구성한다.

4. 결 론

최근 지식기반의 미래사회의 급속한 발달로 소프트웨어 중심의 문제해결력 향상에 관한 관심이 집중되고 이에 관한 교육 및 교육과정의 필요성이 주장되면서 소프트웨어 교육의 관심이 집중되고 있다. 이에 우리정부에서도 소프트웨어 교육의 중요성을 인지하여 2009개정 '정보' 교육과정 이후 2015년 '소프트웨어 교육 운영지침'을 공표했다[7]. 이는 기존의 정보 교육과정의 이론 및 개념 위주의 교육과정에서 탈피하여 Computational Thinking 기반 문제해결방법과 전략을 세우기 위한 알고리즘 설계, 구현에 중점을 두고 있다. 즉 과거 소프트웨어교육이 특정학문분야에 국한된 교육이라면, 현재의 소프트웨어 교육은 미래를 살아가는 우리 모두에게 필요한 필수교육으로 변화하고 있다. 그러나 현재 정부에서 발표한 소프트웨어 교육 운영지침의 교수-학습방법에는 Computational Thinking의 주요개념의 적용방법 및 활용부분에 대한 언급이 부족한 상황이다. 본 연구에서는 소프트웨어 교육 운영지침의 교수-학습방법에 Computational Thinking의 주요개념을 평가요소로 반영하여 소프트웨어 교육의 실질적 방법론의 현실화를 더 하고자 한다.

참 고 문 헌

[1] Bar, D., Harrison, J. & Conery, L.,(2011). Computational thinking: A Digital Age Skill for Everyone. Learning & Learning with Technology, 38 (6), 20-23.

[2] Bundy, A. (2007). Computational thinking is Pervasive. Journal of Scientific and Practical Computing, 1(2), 67-69.

[3] Cuny, J. Snyder, L. & Wing, J. M. (2010). Demystifying Computational thinking for Non-Computer Scientists. Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.

[4] Computer Science Teachers Association & International Society for Technology in Education (2011). Computational Thinking Teacher Resources, http://csta.acm.org/Curriculum/sub/CompThinking.html.

[5] Papert, S. (1996). An Exploration in the Space of Mathematics Educations. International Journal of Computers for Mathematical Learning, Vol. 1, No. 1, pp. 95-123.

[6] Wing, J. M. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35.

[7] 교육부(2015). 소프트웨어 교육 운영 지침.

[8] 권정인(2014). Computational Thinking기반의 교수학습이 학습자의 창의적 문제해결에 미치는 효과성 연구, 성균관대 박사학위논문.

[9] 김경훈외(2012). 미래 한국인의 핵심역량 증진을 위한 창의적 문제해결력 기반의 정보 교육 정책 방향 탐색. 한국교육과정 평가원, 연구보고 RRC 2012-7

◐ 저 자 소 개 ◑

권 정 인

2009년 홍익대학교 교육대학원 컴퓨터교육 석사
2014년 성균관대학교 일반대학원 컴퓨터교육 박사
2015년~현재 성균관대학교 소프트웨어학과 초빙교수
관심분야 : Computational Thinking, SW교육, 인터넷윤리

김 재 현

1988년 성균관대학교 수학과 졸업(학사)
1992년 웨스턴일리노이 주립대학교 대학원 전산학과 석사
2000년 일리노이공과대학교 대학원 전산학과 박사
2014년~현재 한국컴퓨터교육학회 부회장
2010년~현재 한국인터넷정보학회 부회장
2002년~현재 성균관대학교 컴퓨터교육과 교수
2016년~현재 성균관대학교 성균SW교육원 원장
관심분야 : 객체지향 소프트웨어공학, 컴퓨터교육, Computer Based LET etc.