

컴퓨팅 사고력 기반 프로그래밍 학습의 효과성 분석

김경규[†] · 이종연^{††}

요 약

디지털 정보시대가 필요로 하는 역량은 정보를 기반으로 한 문제해결력, 논리적 사고력, 그리고 융합적 사고력 등이다. 이러한 능력을 갖추기 위해 21세기를 살아가는 현대인들은 컴퓨팅 사고(Computational Thinking)를 함양하기 위한 교육을 해야 한다. 그러나 현재 중학교 컴퓨터 교육과정 내에서 이러한 사고력을 향상시키기란 현실적으로 매우 부족한 실정이다. 이에 단편적이고 제한적인 컴퓨터 교육에서 벗어나 학습자의 컴퓨팅 사고능력을 신장시키고 컴퓨터 과학 원리를 보다 쉽게 이해시키기 위하여 프로그래밍 중심의 교육 프로그램을 개발하였다. 개발한 교육 프로그램의 효과를 검증하기 위해 소프트웨어 교육 시범학교로 선정된 W중학교 1,2학년 학생 22명, 3학년 학생 20명을 대상으로 12주간에 걸쳐 총 24차시 동안 적용한 결과, 논리적 사고력과 창의적 문제해결능력 향상의 긍정적 변화를 가져오는 것을 확인하였다. 결과적으로 본 연구에서는 컴퓨팅 사고력 기반 프로그래밍 학습의 현장 적용 가능성을 제시하고 그 효과성에 대해서 분석하였다. 향후 컴퓨팅 사고력을 향상시키기 위한 다양한 교육 프로그램의 개발 방향을 제시하는데 의의를 찾을 수 있다.

주제어 : 컴퓨팅 사고력, 컴퓨터 교육, 소프트웨어 교육, 논리적 사고력

Analysis of the Effectiveness of Computational Thinking-Based Programming Learning

KyungKyu Kim[†] · JongYun Lee^{††}

ABSTRACT

It is necessary for us to be equipped with problem-solving ability, logical thinking ability, and convergence thinking ability in the digital information age. To have these abilities, modern people in the 21st century should be educated to develop the computational thinking ability. However, it is difficult to cultivate the computational thinking ability in current computer education curriculum. Therefore, this paper aims to propose a programming-based computer educational program to make students better understand the principles of computer science and enhance student's computational thinking ability escaping from current fragmentary and limited computer education. In order to verify the effectiveness of the educational program, students in total 42 middle school (22 students in 1st and 2nd grades and 20 in 3rd grade) were applied to the educational program for twenty-four hours on 12 weeks and then we obtained a positive result that might bring improvement on the logical thinking ability and creative problem-solving skills. As a result, this study presents the possibility of its field applications of computational thinking-based programming learning and analyzed the effectiveness. therefore, it has a notable point of presenting the development direction of a variety of education programs for enhancing the computational thinking in the future.

Keywords : Computational thinking, Computer education, Software education, Logical

† 정 회 원: 충북대학교 대학원 컴퓨터학과 박사과정
 †† 중신회원: 충북대학교 소프트웨어학과/컴퓨터교육과 교수(교신저자)
 논문접수: 2015년 11월 13일, 심사완료: 2015년 12월 13일, 게재확정: 2016년 1월 9일

1. 서론

디지털 정보시대가 필요로 하는 역량은 정보를 기반으로 한 문제해결력, 논리적 사고력, 그리고 융합적 사고력 등이다. 이에 의료, 산업, 학문 등 전 분야에서 컴퓨팅 사고력의 필요성을 시사하고 있는바 정보과학에 대한 관심과 투자는 이제 선택이 아닌 필수적인 요소가 되었다. 컴퓨팅 사고력(Computational Thinking, CT)란 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력이다[1]. 이를 기반으로 한 컴퓨터 교육을 강조하는 또 다른 이유는 코딩(Coding) 즉, 컴퓨터 작업의 흐름에 따라 프로그램의 명령문을 사용하여 프로그램을 작성하는 일을 하면 자연스럽게 논리적 사고력과 문제해결력을 기르게 된다는 점에 있다[2].

현재 미국이나 영국, 이스라엘, 유럽 등 해외 여러 나라는 컴퓨팅 능력의 중요성을 인식하고 CT를 교육과정에 도입하기 위하여 다양한 노력을 기울이고 있다. 미국은 새로운 국가수준의 과학교육과정 표준을 연구할 때 과학적 개념과 원리 중심의 학습에서 개념과 원리를 활용한 모델을 구현하는 공학적 실천으로 교육의 방향을 확장하고 있으며, 이러한 변화 과정에서 CT를 활용한 공학적인 실천에 관련된 내용을 교육과정에 포함하고 있다. 또한 컴퓨터 과학이 미국의 STEM(Science, Technology, Engineering, & Mathematics) 교육에서 추구하는 교육 목표에 반드시 필요한 부분이 되면서 컴퓨터과학 교육의 중요한 요소인 CT가 STEM 분야의 핵심이 되어가고 있다[3]. 세계적인 흐름에 맞춰 정보·컴퓨터 교육(이하 ‘컴퓨터 교육’이라 칭함)은 창의적인 문제해결능력을 기르기 위하여 컴퓨터를 활용한 창의적 사고력, 논리적 사고력 신장 즉, 컴퓨팅 사고력에 그 주된 목적을 두고 있다. 이에 따라 2009 개정 교육과정의 컴퓨터 교과는 정보 과학 기술의 기본 개념과 원리를 이해하고, 실생활의 다양한 문제를 컴퓨팅 사고로 관찰하고 해결하는 능력과 정보 윤리적 소양을 기르는데 중점을 두고 있다. 결국 현재의 정보교육의 지향점은 창의적 사고력과 컴퓨팅 사고력 신장에 있다고 할 수 있다[3]. 그러나 컴퓨터 교육의 지향점과 현

장의 컴퓨터 교과 교육내용에는 큰 차이가 존재하고 있다. 초·중등정보통신기술교육 운영지침에 근거한 교육내용을 보면 단순한 정보수집 및 컴퓨터 활용 능력 신장에 집중되어 있고 논리적 사고력과 컴퓨팅 사고기반의 문제해결능력 향상을 위한 컴퓨터교육은 미비한 실정이다[5].

현재 초·중등 교육과정의 컴퓨터 교육내용은 컴퓨터 과학과 관련된 원리를 단편적으로 제시할 뿐 컴퓨팅 사고력을 신장시키기에는 부족하다. 또한 응용프로그래밍 활용 교육에 집중되어 있어 컴퓨터 동작 원리를 기반으로 한 컴퓨팅 사고력과 창의적 문제해결능력의 신장을 위한 체계적 교육은 이루어지지 않고 있는 실정이다[6]. 이에 컴퓨터 교육은 소프트웨어, 컴퓨터시스템, 융합활동 등 학문의 기본적인 원리와 개념을 갖고 있는 학문으로 특히 알고리즘과 로직의 설계, 프로그래밍 등을 통한 컴퓨터 기반 문제해결뿐 아니라 컴퓨팅 사고력을 함양하며, 일반교과별 특성에 맞는 컴퓨팅 사고력 기반 교육 프로그램 개발이 필요하다. 또한 컴퓨터 교육의 핵심인 컴퓨터 과학 원리를 친근하고 다양한 문제 상황을 통해 경험하고 이해 할 수 있는 교육내용이 제공되어야 한다.

본 연구에서는 프로그래밍 중심 소프트웨어 교육을 현장 적용을 통해 일선 현장에 적용 가능한 교육의 방향성을 제공하고 그 교육적 효과를 검증하는 과정을 거친 후 개선점과 운영방안을 제 공함으로써 소프트웨어 교육 확산에 기여하고자 한다.

2. 관련 연구

2.1 컴퓨팅 사고력

Wing(2008)은 컴퓨팅 사고가 3R(읽기, 쓰기, 셈하기)과 더불어 모든 학습자가 갖추어야 할 기본 능력이며, 추상화 (Abstraction)와 자동화 (Automation)를 통한 문제해결능력이라고 하였다. 컴퓨팅 사고가 문제 해결 과정에서 추상화 과정을 통해 문제의 핵심 요소를 추출하고 모델링하여 컴퓨팅 기기를 통해 해법을 자동화하는 능력을 의미한다고 하였다. 또한 단순히 컴퓨터 과학 및 프로그래밍에 대한 지식을 가지고 있거나

프로그래밍을 실천하는 기술적인 능력을 가지고 있는 것만이 컴퓨팅 사고능력을 가지고 있는 것이 아니라, 문제의 인식과 분석, 자료 수집과 분석, 문제 해결책 마련을 위하여 다양한 사고를 진행하고, 그 과정에서 컴퓨팅 능력을 활용하고 그 해결책을 정보과학적으로 구현하는 방법을 설계하는 과정 전반을 컴퓨팅 사고로 생각하고 있다 [7]. 그리고 컴퓨팅 사고의 정의에 대한 논의는 워싱턴 D. C에서 2009년 2월과 2010년 2월에 열린 두 차례의 워크숍에서 공식적으로 논의되었다. 관련 논의는 CT의 핵심적 개념에 대한 논의, CT의 구조 및 개인의 CT 능력의 측정, CT 교육 방안 등에 대해 진행되었다. 이 워크숍에서 논의된 CT의 정의는 학자마다 조금씩 다르며 이를 정리하면 <표 1>과 같다[8].

<표 1> Computational Thinking에 대한 학자들의 정의[8]

학자	Computational Thinking 정의
David Moursund	Saymour Papert가 그의 저서 'Mindstorms' 에서 언급 하였던 절차적 사고와 연관되어 있는 개념이다. 절차적 사고는 개발(developing) - 표현(representing) - 시험(testing) - 디버깅(debugging)의 순서들로 구성되어 있고, 효과적인 절차를 만들기 위해서는 컴퓨터 등의 특별한 장치로 수행 가능한 형태인 구체적인 단계별 명령이 필요하다.
Peter Lee	CT는 인간 지능을 확대하여 실제적인 적용을 할 수 있는 인간지능의 메커니즘에 관한 연구이다. 다시 말해 인간의 정신적인능력의 복잡도를 관리하거나 일을 자동적으로 처리하도록 하는 추상화 도구를 통해 확장하는 것이라 할 수 있다.
Bill Wulf	과학은 물리적인 대상(physical object)에 관련된 것이고, CT는 어떠한 문제를 해결하는 과정과 그 과정의 진행을 가능하게 하는 추상적인 현상들에 초점을 맞추고 있다.
Don Abrahamson	CT는 형식지를 설명하고, 암묵지를 객관화하고, 이러한 지식을 컴퓨팅적인 형태로 퍼뜨리고, 이러한 활동에서 발생한 결과물을 관리하는 컴퓨터 관련 기호 체계의 사용이다.
Gerald Sussman	CT는 일을 처리하는 정확한 방법을 공식화하는 방법이다. 다시 말해 특정한 문제를 효율적으로 처리하기 위해 그 문제를 철저하게 분석하고 해결하기 위한 엄밀한 절차를 만드는 과정이 CT인 것이다.
Edward Fox	문제 해결을 목적으로 눈에 보이지 않는 추상화된 개념을 다루거나 조작하는 것이 CT의 핵심이다. CT는 "인간이 세상에 접근하고, 과정들을 생각하고, 디지털로 표현된 것들을 다룰 때 하는 일들" 이라고 표현할 수 있다.
Robert Constable	CT는 특정한 기술이나 사고 과정들의 집합이 아니라 개방적이고 점점 발전하는 기술의 역동적인 본질에 관련된 개념이다. 현재까지 나타난 CT를 대략적으로 기술한다면 "지적인 과정을 자동화하는 것" 과 "정보 처리 과정을 연구하는 것" 정도가 될 것이다. CT가 특별하게 받아들여지는 이유는 우리가 생각하는 "컴퓨팅 사고력" 이 컴퓨터라는 도구를 통해 실행될 수 있다는 점이고, 이러한 점은 "컴퓨터가 파트너 및 동지가 될 수 있다." 는 발견을 가져왔다는 것이다.

따라서 여러 학자들의 의견과 선행 연구들을 바탕으로 컴퓨팅 사고력 도입이 학습자들에게 가

져올 수 있는 교육적 효과는 다음과 같다[9].

첫째, 의사소통과 협업을 촉진시킬 수 있다. 컴퓨팅 사고력은 인간의 사고를 직접적으로 다루고 있으며, 실생활의 문제를 탐색하고 해결책을 모색하는 과정에서 문제 해결을 위한 도구로 컴퓨팅 사고력을 활용하게 되면서 타인과의 협업, 의사소통이 이루어지도록 한다.

둘째, 문제 해결에 있어 적절한 도구와 전략을 선택하고 활용하도록 한다. 컴퓨팅 사고력 교육의 도입은 학습자로 하여금 컴퓨팅 사고력을 개발하고 활용할 경험을 제공하게 된다. 컴퓨팅 사고력은 문제 해결을 위한 적절한 전략을 수립하고 이 전략을 바탕으로 컴퓨팅 시스템이 문제를 해결하도록 적절한 자동화 도구를 사용하는 것을 포함하기 때문에 실생활이나 다양한 학문 분야에서의 문제 해결에 있어 효율성을 향상시킬 수 있다.

셋째, 학습자들이 정보화 사회, 디지털 세상에 잘 적응할 수 있도록 한다. 컴퓨팅 사고력 교육은 현대사회에서 컴퓨팅 사고력의 역할을 이해하게 하고, 습득한 배경 지식들을 적재적소에 어떤 방식으로 효과적으로 사용하도록 할 것인지, 어떤 지식을 갖추고 있어야 하는지 그 필요성과 중요성을 깨닫도록 한다.

넷째, 컴퓨팅 사고력의 향상은 곧 문제 해결 능력의 향상이다. 컴퓨팅 사고력 교육의 도입은 필요한 정보를 효과적으로 수집하는 능력, 수집한 자료들의 패턴을 분석하는 능력, 문제 해결에 필요한 필수 요소를 걸러내는 능력, 문제 해결 과정을 절차적으로 설계하는 능력, 문제 해결을 위하여 컴퓨팅 시스템을 활용하는 능력을 기르도록 함으로써 궁극적으로는 학습자들의 창의적 문제 해결능력을 향상시키도록 한다.

다섯째, 학습의 향상을 가져다준다. 컴퓨팅 사고력은 기초 학문 분야에서 학습한 개념과 원리를 공학적 측면에서 실천하도록 하여 학습자로 하여금 학습한 지식들이 왜 필요한지, 어떻게 만들어졌는지를 이해하도록 하며, 단편적인 개념이나 원리를 서로 연계시켜주거나 인간의 사고가 갖는 한계에 의해 발생하는 사고의 오류들을 수정해줌으로써 인지 구조의 확장이 가능하도록 한다.

이에 컴퓨팅 사고력 학습은 복잡하고 모호한

문제를 분석해 손에 잡히는 구체적인 문제로 바꾸고, 이를 해결하기 위해 필요한 데이터를 찾아 모델을 만들어 답을 얻어내는 과정을 말한다. 그러므로 학생들에게 창의적 사고능력과 논리적 사고력을 키우는데 가장 적합하다는 것이다.

2.2 프로그래밍과 논리적 사고력

논리적 사고력은 비판적 사고의 좁은 의미로 추리능력을 의미하며, 비판적 사고는 어떤 주장이나 정보에 대한 타당한 가치를 판단하기 위해 객관적으로 분석하는 능력으로 논리적 사고의 포괄적인 개념으로 볼 수 있다. 결국 논리적 사고력은 사상(event)들 간의 관계, 모순 등 일련의 규칙을 타당성(validity)에 준거를 두고 사고하는 추리능력이다. 논리적 사고의 하위요소는 Piaget이론에 기반을 둔 표준 논리 검사 요소인 계열화 논리, 비례 논리, 확률논리, 변인통제논리, 조합논리, 명제논리 등 총 6가지를 말한다. 이들 하위요소는 <표 2>와 같은 능력을 의미한다[10].

<표 2> 논리적 사고의 하위요소[10]

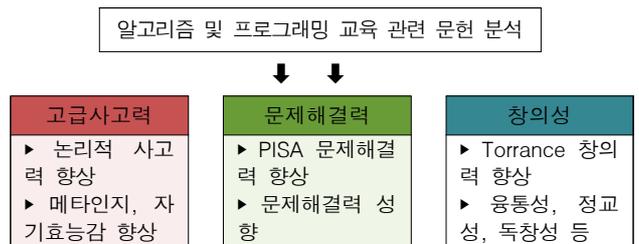
하위 요소	내용
계열화 논리	어떤 사물의 크기, 무게, 부피의 증감에 따라 일련의 요소들을 규칙에 따라 정신적으로 배열하는 능력
비례논리	두 비에 있어서 그 비의 값이 같다는 논리를 바탕으로 비례와 관련된 규칙 혹은 관계들 간의 관계(Relation between relations)를 이해하는 능력
조합논리	사건이 일어날 수 있는 모든 경우를 빠짐없이, 중복되지 않도록 따져 보는 능력
명제논리	명제는 참인지 거짓인지를 원칙적으로 분명히 판별할 수 있는 문장을 말할 때, 명제논리는 명제와 명제의 결합 및 그 결합의 구조를 체계적으로 분석하는 능력
변인통제 논리	어떤 문제에 직면했을 때, 상황의 모든 변인들을 인식하고 변인들의 역할에 관한 가설을 설정한 다음, 그 가설을 검증하기 위해 체계적으로 변인들을 통제하여 결론을 도출해내는 능력
확률논리	여러 가지 현상 속에 숨어있는 규칙을 파악하여 미래를 예측하고 판단할 수 있는 능력

따라서, 논리적 사고력은 컴퓨팅 환경에서 문제 해결의 절차를 마련하거나 문제해결방법 및 과정의 옳고 그름을 판단하기 위해 필요한 사고능력 중 하나이다[11]. 문제를 해결하는 하나하나의 과정을 나열하면서 그에 따른 결과 값이 어떻게 되고 그 결과 값을 다시 사용하여 새로운 결과 값을 도출하며 최종적으로 내가 원하는 결과 값까지 도출될 수 있는지 확인하고 구조화 시키는 과

정에서도 논리적 사고력이 요구된다. 특히 문제를 해결하는 방안으로써 프로그래밍 과정은 절차적이고 논리적인 과정으로 이루어지기 때문에 논리적 사고력이 많이 요구된다. 이에 프로그래밍 언어들은 논리의 기본적인 개념에 기초를 두고 있기 때문에 다양한 인지 기술이 복합된 과정으로 논리적 사고 능력과 추상적인 추론 능력이 요구되어 학습자의 인지발달에 영향을 미치고 어떤 프로그래밍 언어라도 프로그램 자체가 논리이므로 이러한 논리적인 요소를 내포하고 있어서 논리적 사고력 형성에 어떤 식으로든 영향을 준다.

2.3 교육용 프로그래밍 언어의 교육적 효과

교육용 프로그래밍 언어(Educational Programming Language, EPL)를 활용한 초·중등 교육의 프로그래밍 교육은 다양한 교육적 효과로 나타난다. 대표적으로 <그림 1>과 같이 논리적 사고력, 창의성, 문제해결력과 같은 학습자의 능력이 증대되며 스크래치, 두리틀, LOGO 등과 같은 언어를 사용할 경우 학습자의 흥미와 재미를 만족시키는 효과들이 연구되었다[12].



<그림 1> EPL 적용 교육적 효과[12]

EPL을 적용한 프로그래밍 수업에서 나타난 교육적 효과를 분석해 보면 <표 3>과 같이 문제해결력, 창의성, 자기효능감, 수업 몰입, 학습동기 등이 향상된 것을 알 수 있다[13].

<표 3> EPL 사용 수업 후 교육적 효과 분석

연구자	제목	연구결과
이은경, 이영준 (2008)	스크래치 활용 프로그래밍 교육이 중학생의 몰입수준과 프로그래밍 능력에 미치는 영향	<ul style="list-style-type: none"> 스크래치와 비주얼베이직 비교 중학교 1학년 몰입 및 프로그래밍 능력 향상
조성환, 송정범, 김성식, 백성혜 (2008)	스크래치를 이용한 프로그래밍 수업 효과	<ul style="list-style-type: none"> 중학교 1,2학년 30명 메타인지 향상, 여자집단이 더 향상 자기효능감 향상, 여자집단은 무의미

이점순 (2008)	LOGO 프로그래밍 언어가 초등학생 의 창의성 발달에 미치는 영향	<ul style="list-style-type: none"> 로고 사용, 초등학교 4학년 2개반 실험반이 창의성 향상됨
배학진, 이은경, 이영준 (2009)	문제 중심 학습을 적용한 스크래치 프로그래밍 교수 학습 모형	<ul style="list-style-type: none"> 스크래치 사용, 초등학교 5학년 2개반 문제해결력 성향 향상 논리적 사고력 향상
이민희 (2009)	두리틀을 이용한 프로그래밍 수업이 의성, 문제해결력, 로고 프로그래밍 흥미도 향상에 미치는 영향.	<ul style="list-style-type: none"> 두리틀 사용 초등학교 5학년 34명 두리틀 프로그래밍과 기존컴퓨터수업 비교 창의성유의미, 문제해결력 유의미, 프로그래밍 흥미도 사전사후 유의미
서영민 (2010)	초등정보영재의 창의성 신장을 위한 교과통합 로봇 프로그래밍 수업 모형	<ul style="list-style-type: none"> 로봇프로그래밍, 초등정보영재 대상 32명 교과통합 로봇프로그래밍과 일반적 로봇프로그래밍 비교 창의성성향 유의미(개방성, 호기심 높음), 창의성인지 유의미
송정범, 조성환, 이태욱 (2008)	스크래치 프로그래밍 학습이 학습자의 동기와 문제해결력에 미치는 영향	<ul style="list-style-type: none"> 스크래치와 비주얼베이직 비교(6학년 67명) 스크래치가 학습동기에 영향, 문제해결력이 향상
조성환, 송정범, 김성식, 이경화 (2008)	CPS에 기반한 스크래치 EPL이 문제해결력과 프로그래밍 태도에 미치는 효과	<ul style="list-style-type: none"> 스크래치 사용(중학교 1학년 28명) 문제해결력 향상, 프로그래밍 태도 향상
정미연, 이은경, 이영준 (2008)	Squeak Etoys 활용 알고리즘 학습이 중학생의 문제해결력에 미치는 영향	<ul style="list-style-type: none"> 스퀘사용과 이론중심 비교(중학생 2개반) 문제해결성향 향상 논리적사고력 실험, 통제 동시 유의미, 사후검사이 두 집단 차이 유의미, 자기효능감 향상
송정범, 백성혜, 이태욱 (2009)	성별의 차이를 고려한 로봇 프로그래밍 학습이 여중학생의 몰입수준과 문제해결력에 미치는 효과	<ul style="list-style-type: none"> 피코크리켓 활용 로봇 프로그래밍(중학교 2학년) 몰입수준 사후결과 향상, 성별로는 사전, 사후 차이가 없다 문제해결력 사전 사후 유의미
구정모 외 (2009)	문제 중심 스토리텔링 프로그래밍 학습이 학습동기 및 문제해결능력에 미치는 효과	<ul style="list-style-type: none"> 앨리스 사용, 일반적인 프로그래밍 수업과 문제 중심 스토리텔링 비교 초등학교 5-6학년 학습동기 유의미, 문제해결능력 유의미
이은경 (2009)	문제해결력 향상을 위한 과제 중심 스크래치 프로그래밍 학습 프로그램	<ul style="list-style-type: none"> 스크래치 사용, 중학교 1학년 문제해결력에서는 과제중심>스크래치>비주얼베이직 유의미, 의사결정과 문제점 해결영역 유의미, 세 집단 모두 문제해결력 향상

진영학, 김영식(2011)은 EPL을 사용한 교육에 대한 교육효과를 검증하기 위해 메타분석을 실시하고 그 효과를 검증하는데, 위에서 분석한 각 연구의 결과와 일치하게 학업성취도, 논리적 사고, 문제해결력, 창의성, 학습흥미 등에 효과가 있다는 사실을 분석하였다[14].

따라서 위의 여러 사례연구들에서 교육적 효과가 검증된 바 컴퓨팅 사고를 키울 수 있는 프로그래밍 교육은 EPL 학습 도구를 활용하면 효과적이라고 말할 수 있다.

3. CT 기반 교육 프로그램의 설계

일선 현장에 적용 가능한 프로그래밍 중심의 교육 프로그램 개발에 있어 컴퓨팅 사고력 도입

이 가능한 적절한 학습 주제 선정을 비롯한 이에 적절한 교수·학습 전략 및 학습 도구의 선정 후 개발된 교육내용을 적용하기 위하여 <표 4>와 같이 계획, 준비, 개발, 검토, 적용 및 분석의 5단계를 설정하고 6개월 동안 각 단계에 따라 연구를 진행하여 CT 기반 프로그래밍 교육 프로그램을 개발하였다.

<표 4> 연구 설계 단계

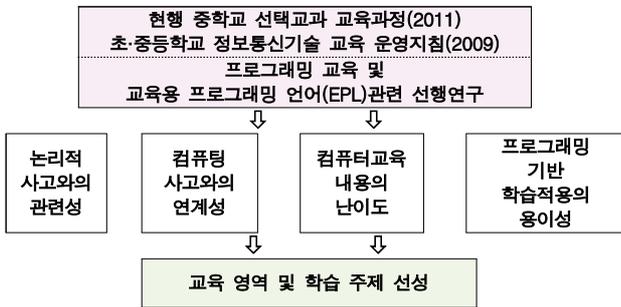
계획 단계	<ul style="list-style-type: none"> 기존 교육과정, 관련문헌 및 선행연구를 통한 컴퓨터과학 영역별 CT기반 소프트웨어 교육관련 주제 선정
준비 단계	<ul style="list-style-type: none"> 계획 단계에서의 분석을 토대로 CT 기반 소프트웨어 교육 프로그램 운영을 위한 교수·학습 전략 연구 검사도구 선정
개발 단계	<ul style="list-style-type: none"> 컴퓨터과학 세부 교육 영역별 SW 교육 프로그램 개발
검토 단계	<ul style="list-style-type: none"> 전문가 검토 및 피드백 제공 CT기반 교수·학습 전략 방법 수정 및 보완
적용 및 분석 단계	<ul style="list-style-type: none"> CT기반 소프트웨어 교육 프로그램 적용 결과분석 및 효과 검증

3.1 교육영역 및 학습 주제 선정

CT는 컴퓨터 과학과 기술을 기반으로 급변하고 있는 사회에서 학습자들이 갖추어야 할 기본적인 사고 능력으로 대두되고 있는 상황이다. 이에 본 교육영역 및 학습주제를 단순한 프로그래밍 언어, 컴퓨터 과학 개념 전달 보다 컴퓨터 과학의 개념과 프로그래밍을 통한 ‘문제해결능력과 논리적 사고력 향상’에 초점을 두었다. 또한 프로그래밍 학습은 학생을 컴퓨터 과학자나 프로그래머로 만들기 위함이 아니라, 컴퓨팅 사고력을 통한 창의적 문제해결능력과 논리적 사고력 향상시키기 위함이다.

CT의 학습목표에 대한 선도적 연구자들의 견해를 살펴보면 다음과 같이 다섯 가지의 항목으로 간추릴 수 있다[15]. 첫째, 추상적 기능의 내면화이다. 둘째, 자기 표현력과 세계에 대한 컴퓨팅 사고적 이해력이다. 셋째, 다양한 분야에서 컴퓨팅 사고 개념의 활용력이다. 넷째, 변화에 대한 적응력이다. 다섯째, 사회적 협동심 함양이다. 따라서 본 교육 프로그램의 교육내용은 컴퓨팅 사고를 키우는데 중점을 두고 있기에 다음과 같이 구성하고자 한다. 첫째, 필요한 자료를 수집하고

자료를 분석하여 문제를 해결할 수 있는 단위로 분해해 본다. 둘째, 복잡한 문제를 단순화하고 문제를 해결하는 절차를 정리하여 현재 제시된 문제를 해결하기 위한 절차를 설계해 본다. 셋째, 절차에 따라 컴퓨팅 기기에서 자동화하여 실행될 수 있도록 코딩을 수행해 본다. 코딩은 문제 해결 과정에 따라 자신의 생각을 컴퓨팅 기기가 자동화하여 실행 할 수 있도록 만드는 명령이라는 것을 강조한다. 넷째, 완성된 교육내용은 자신의 컴퓨팅 기기에서 직접 실행하여 시뮬레이션 해 본다. 실행한 결과에 대해 스스로 평가해 보고 개선해야 할 점에 대해 새로운 아이디어를 구현해 보도록 한다.



<그림 2> 컴퓨터 교육영역 및 학습 주제 선정 과정

<표 5> 교육영역 및 내용체계

교육영역	교육 내용체계
프로그래밍 개념과 원리	<ul style="list-style-type: none"> ▶컴퓨터과학과 소프트웨어의 이해 ▶코드닷오알지와 엔트리를 활용한 프로그래밍 개념과 원리 학습(순차, 반복, 선택, 변수, 함수) ▶프로그램 작성법 및 디버깅 ▶일반 프로그램 만들기
알고리즘 및 프로그래밍	<ul style="list-style-type: none"> ▶알고리즘 정의 ▶알고리즘 표현 방법 ▶Raptor를 활용한 알고리즘 설계 및 실행 <ul style="list-style-type: none"> • 프로그램 순서도와 의사코드 작성하기 ▶앱 인벤터를 활용한 모바일 게임 개발 <ul style="list-style-type: none"> • 문제해결 • 알고리즘 설계 • 프로그래밍 • 오류 수정
컴퓨팅과 문제 해결	<ul style="list-style-type: none"> ▶Mindstorm을 활용한 로봇 프로그래밍 ▶센서와 코딩을 통한 로봇 원리 이해 ▶Greenfoot을 통해 객체지향 프로그램 이해 ▶객체지향 프로그램 분석과 설계 <ul style="list-style-type: none"> • Greenfoot을 활용한 슈팅게임 만들기
정보윤리	<ul style="list-style-type: none"> ▶개인정보보호 및 정보 보안 ▶지적 재산의 보호

따라서 <그림 2>와 같이 교육과학기술부(現 교육부)에서 제시한 현행 중학교 선택교과 교육과정

(2011)[4], 초·중등학교 정보통신기술 교육 운영지침(2009)[5], 프로그래밍 교육 및 EPL관련 선행연구들을 분석하여 위의 견해와 구성을 토대로 교육영역을 고르게 포함하면서 투입 가능한 영역과 주제를 선별하여 <표 5>와 같이 영역별 교육내용체계를 구성하였다. 또한 추출된 학습주제를 바탕으로 컴퓨팅 사고와의 관련성, 논리적 사고와의 관련성, 컴퓨터과학 교육 내용의 난이도, 프로그래밍 학습 적용의 용이성 등을 고려하여 중학생을 대상으로 교육 가능한 세부 교육 주제를 선정하였다.

3.2 사고 영역 선정

컴퓨팅 사고력은 International Society for Technology in Education(ISTE)와 Computer Science Teachers Association(CSTA)에서 제시한 자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화의 9가지의 세부요소<표 6>(ISTE & CSTA, 2011)로 사고영역 중심의 학습내용을 구성하였다 [16].

<표 6> CT의 세부 구성요소

구성요소	정의
자료수집	해결해야 하는 문제와 관련된 알맞은 자료를 모으는 과정
자료분석	자료를 이해하고, 패턴을 찾아 결론을 도출
자료표현	적절한 그래프, 차트, 글, 그림 등으로 자료 정리
문제분해	문제를 해결가능한 수준의 작은 문제로 나누기
추상화	문제 해결을 위해 반드시 필요한 핵심 요소를 파악하고, 복잡함을 단순화
알고리즘 및 절차	문제를 해결하거나 어떤 목표를 달성하기 위해 수행되는 일련의 단계
자동화	컴퓨팅 시스템이 수행할 수 있는 형태로 해결책 나타내기
시뮬레이션	자동화의 결과이며, 문제를 해결하기 위하여 만든 모델을 실행시켜 결과 파악하기
병렬화	목표를 달성하기 위한 작업을 동시에 수행하도록 자원을 구성

이에 개발한 프로그래밍교육 프로그램 교육내용에서는 단순히 컴퓨터과학 내용을 이해시키는 데 목적이 있는 것이 아니라 이를 통해서 효과적인 문제해결 과정을 경험함으로써 컴퓨팅 사고를 통한 논리적 사고력과 문제해결능력 신장을 목표로 한다. 그리고 동일한 소재를 갖고 동일 컴퓨터과학 내용을 지도하더라도 CT세부요소가 무엇인

<표 7> 학습 주제별 학습내용 및 CT 중점 요소

차시	학습주제	학습목표	주제별 학습내용	컴퓨팅 사고(Computation Thinking)	
				학습 요소	CT 중점 요소
1	컴퓨터 과학 과 소프트웨어	소프트웨어의 개념과 종류를 이해하고 설명할 수 있다.	<ul style="list-style-type: none"> 컴퓨터과학과 소프트웨어 소개 일상생활 속의 소프트웨어 알아보기 소프트웨어 개발 방법과 환경 		
2	프로그래밍의 이해	프로그래밍의 기본 개념을 이해하고 설명할 수 있다.	<ul style="list-style-type: none"> 프로그래밍 기본 개념과 원리 엔트리 보드게임을 통해 프로그래밍 기본 원리 학습하기 	분제의 분석과 표현	자료수집 자료표현 자료분석 알고리즘 및 절차
3-5	프로그램 작성법 및 디버깅	프로그래밍 개발 환경과 기본 기능들을 익히고 사용할 수 있다.	<ul style="list-style-type: none"> 코드닷오알지/엔트리 소개 코드닷오알지를 학습모드를 통한 프로그램 작성법 및 디버깅 알기 엔트리를 통해 변수의 개념과 활용하기 엔트리를 통해 순차, 선택, 반복구조 이해하기 	변수의 개념과 활용 자료의 입력과 출력 제어문의 이해와 표현	
6-7	애니메이션 만들기	애니메이션 원리를 알고 엔트리를 활용하여 프로그래밍 할 수 있다.	<ul style="list-style-type: none"> 애니메이션 원리를 이해하고 애니메이션 기본 예제 따라 하기 나만의 애니메이션(이야기책) 만들기 	프로그램 분석과 표현 시뮬레이션의 적용	
8-9	일반 프로그램 만들기	간단한 일반 프로그램을 작성할 수 있다.	<ul style="list-style-type: none"> 예제 일반프로그램을 분석하고 프로그램 작성하기 - 사진첩 프로그램 만들기 - 투표 프로그램 만들기 	프로그램 분석과 표현 프로그램 해결과정 시뮬레이션의 적용	자료수집 자료표현 자료분석 추상화 자동화
10	알고리즘의 표현과 이해	알고리즘의 표현방법을 이해하고 표현할 수 있다.	<ul style="list-style-type: none"> 알고리즘 정의와 표현 방법 알고리즘의 다양성과 효율성 RAPTOR를 활용한 프로그램 순서도 작성하기 	알고리즘 정의 알고리즘 설계	자료수집 자료표현 자료분석 알고리즘 및 절차 추상화 자동화
11-12	미니 게임 알고리즘	프로그램 순서도를 작성하고 모바일 프로그램을 만들 수 있다.	<ul style="list-style-type: none"> 모바일 미니게임 예제 분석하기 분석한 미니게임 모바일 프로그램 순서도 및 의사코드 작성해보기 	프로그램 분석과 표현 알고리즘 설계 알고리즘 실행	
13-15	모바일 게임 만들기	앱 인벤터를 활용하여 모바일 게임을 만들 수 있다.	<ul style="list-style-type: none"> 앱 인벤터 소개 및 개발 환경 익히기 앱 인벤터를 활용한 두더지잡기 게임 만들기 	추상화 사례 제시 시뮬레이션의 적용	
16-17	피지컬 컴퓨팅	피지컬 컴퓨팅의 개념을 이해하고 설명할 수 있다.	<ul style="list-style-type: none"> 피지컬 컴퓨팅의 개념과 예시 살펴보기 구동모터의 동작 원리를 이해하고 장애물 피하기 	상황 제시 로봇 자료 수집 로봇 자료표현 및 자료분석 문제분해	자료수집 자료표현 자료분석 자동화 시뮬레이션
18-19		마인드스톡을 활용하여 로봇의 원리를 이해하고 미로 탈출 로봇을 설계할 수 있다.	<ul style="list-style-type: none"> 미로 탈출 로봇 만들기 위한 적외선 센서 이해하기 미로 탈출 로봇 설계 및 순서도 작성하기 미로 탈출 로봇 프로그램 구현하기 	알고리즘 설계 알고리즘 실행 시뮬레이션의 적용	
20	객체지향의 이해	객체지향 개념을 이해하고 특징을 설명할 수 있다.	<ul style="list-style-type: none"> 객체지향 개념 객체지향 프로그래밍 특징 객체지향 설계를 기반으로 한 게임 프로그램 분석하기 	프로그램 분석 및 단순화 추상화 사례 제시	자료수집 자료표현 자료분석 추상화 자동화 병렬화
21-22	객체지향 기반의 프로그래밍 설계 및 구현	Greenfoot를 활용하여 간단한 슈팅게임을 만들 수 있다.	<ul style="list-style-type: none"> Greenfoot 개발환경 익히기 Greenfoot을 활용한 객체지향 슈팅게임 프로그램 제작하기 	객체지향 프로그램 분석 알고리즘의 설계 알고리즘의 실행 알고리즘의 평가 일반화	
23-24	프로젝트	나만의 프로그램을 제작할 수 있다.	<ul style="list-style-type: none"> 나의 첫 프로그램 계획 및 설계하기 나의 첫 프로그램 제작 및 확장하기 프로그램 발표 및 평가하기(되돌아보기) 		

지에 따라 교수·학습과정에서 교사 발문의 빈도와 형태, 소재의 투입시간, 설명 방법 등에는 큰 차이가 있을 수 있다. 따라서 세부 구성요소들은 교육 프로그램에 반드시 나열된 순서에 따라 배열되거나, 모든 하위 요소가 교육 과정에 포함되어야 할 필요는 없지만 컴퓨팅 사고력 하위 요소들 간의 고른 발달을 도모하고자 하였다.

3.3 교육내용 구성 및 교수·학습 설계

교육내용 구성은 컴퓨터를 이용한 자료와 정보의 수집 및 분석, 문제해결 절차 설계를 위한 컴퓨팅 사고력 함양, 소프트웨어 기초 소양 함양 및

설계 체험 등의 내용으로 구성하였다. 또한 선정된 세부 교육 주제에 따라 컴퓨팅 사고력 기반의 프로그래밍 학습내용을 설계하기 위해 해당 컴퓨터 과학 원리를 효과적으로 표현할 수 있는 활동을 통한 학습(Learning by Doing), 설계를 통한 학습(Learning by Design), 제작을 통한 학습(Learning by Making)을 위한 알고리즘 및 프로그래밍 학습 위주의 흥미로운 교육 소재를 선정하였다.

교육 소재는 관련 컴퓨터 과학 원리를 효과적으로 표현할 수 있어야 하며 학생들의 참여 동기와 학습지속력을 높이기 위해 쉽게 접근할 수 있

고 프로그래밍을 처음 접하는 중학교 학생들에게 친숙하고 재미있게 접할 수 있는 퍼즐형식의 프로그래밍 학습 도구로 블록 기반의 코드닷오알지, 엔트리, 스크래치, 앱 인벤터, Greenfoot 등과 컴퓨터 외부에 물리적으로 존재하는 것을 프로그램으로 제어해보는 피지컬 컴퓨팅 등을 활용한 CT 기반 사례[9][10][11][13]를 바탕으로 <표 7>과 같이 총 12주 24차시로 구성하였다. 그리고 소프트웨어를 개발하는 프로그래밍 학습은 <표 8> 같이 고도화된 추상화 과정을 포함하고 있기에 이러한 추상화된 과정은 구체물을 통해 가시화할 수 있도록 함으로써 학습 효과를 더욱 높일 수 있기 때문이다.

<표 8> 프로그래밍을 통한 학습의 사고력 경험

프로그래밍 학습	사고력 경험
프로그래밍 학습 과정에서 경험	알고리즘적 사고 재귀적 사고
오류 디버깅 과정에서 경험	논리적 사고 비관적 사고
프로그램 설계 과정에서 경험	창의적 사고 발산적 사고

본 교육 프로그램의 교육목표를 달성하기 위한 영역별 교육내용의 적합성에 따른 내용 타당도 검증은 실시하여 타당도를 높이고, 질적 수준을 향상시키기 위해 교육내용 선정, 교수.학습자료 개발, 학습설계 등의 제작과정에서 해당 분야 교수 및 석사, 박사급 관련 연구자와 컴퓨터교육 및 컴퓨터공학 학사학위 이상으로 초.중.고등학교에서 컴퓨터 교과를 가르친 경험이 있는 교사 및 강사 등 8명으로 구성된 교육전문가 집단과 협의과정을 거쳐 수정.보완 작업하였다.

4. 교육 프로그램 적용 및 결과분석

4.1 연구대상

본 연구 대상은 컴퓨팅 사고력 기반 프로그래밍 학습이 학습자의 문제해결력과 논리적 사고력에 각각 어떠한 영향을 미치는지 알아보기 위해 <표 9>와 같이 미래창조과학부 주최 한국과학창의재단 주관하는 소프트웨어 교육 시범학교로 선정된 W중학교 1,2학년 22명과 3학년 20명 대상으

로 사전·사후검사 실험집단을 설계하였다. 기간은 12주간에 걸쳐 총 24차시의 수업을 시행하였으며, 논리적 사고력과 컴퓨팅 사고기반 문제해결능력 향상의 유의미한 차이가 있는지에 대한 대응표본 *t-test*을 실시하였다. 이는 효과성 분석을 위한 것이다.

<표 9> 연구 대상 집단과 사례 수

항 목	연구대상	N(사례 수)
학 년	1학년	8
	2학년	14
	3학년	20
	계	42

4.2 프로그램 운영기간 설정

본 연구는 논리적 사고력과 컴퓨팅 사고기반의 문제해결능력의 변화를 위해 필요한 최소한의 교육내용 투입기간과 시간을 확인하기 위해 관련된 선행 연구를 살펴보았다. 신갑천(2010)은 스크래치를 활용한 로봇 프로그래밍을 1차시부터 15차시까지 수행한 결과 논리적 사고력 검사를 통해 비례논리 확률논리 조합논리에 효과가 있는 것으로 나타났다[17]. 김태미(2014)는 중학생 15명을 대상으로 8차시에 걸쳐 STEAM 프로그램을 진행하고 창의성의 발현빈도와 빈도대비 점수가 향상되는 것을 확인하였다[18]. 또한 서성원(2010)은 로봇 프로그래밍 교육이 컴퓨팅 사고 능력에 미치는 영향에 대한 연구에서 20차시 실시 후 컴퓨팅 사고능력 향상의 차이를 검증하였다[18]. 이와 같은 선행 연구결과를 토대로 단기간에 집중 교육을 통해서도 학습자의 사고능력의 변화가 가능함을 확인 할 수 있고 이에 12주간 24차시 교육 프로그램 운영을 통해서도 중학교 학생들의 논리적 사고력과 문제해결능력 변화가 가능할 것이라 예상할 수 있었다.

4.3 연구 설계 및 절차

본 연구의 실험설계는 <표 10>과 같이 실험집단을 대상으로 중학교(1~2학년)과 중학교(3학년)으로 나누어 반별로 사전 논리적 사고력 검사와 컴퓨팅 사고기반의 문제해결능력 검사를 각각 실시하였다. 이후 CT 기반 프로그래밍 중심 학습을

9월 2주~12월 2주까지(12주간) 24차시를 운영한 후 동형 검사지를 활용하여 실험집단의 사전·사후 논리적 사고력 및 컴퓨팅 사고기반의 문제해결능력의 향상 정도를 대응표본 *t-test*으로 검사하였다.

<표 10> 실험 설계

실험집단	O1	X1	O2
	O1 : 사전검사 , O2 : 사후검사		
	X1 : CT 기반의 프로그래밍 학습프로그램 운영		

검사도구로 Piaget 이론 기반의 표준 논리 검사 요소인 계열화 논리, 조합 논리, 명제 논리, 확률 논리, 비례 논리, 변인 통제 논리 등 총 6가지 요소에 대해 검사하는 논리적 사고력 검사지[19][20]를 활용하였다. 또한 컴퓨팅 사고기반의 문제해결능력 신장 검사는 Marneffe(1998)와 Lewandowski(1998)의 이론을 기초로 하여 고려대학교 대학원에서 개발한 ‘컴퓨팅 사고 기반의 문제 해결 능력 문항지(중학교)’ 사전·사후 검사지[21]를 이용하였다. 자료 처리 방법은 SPSS(Statistical Package for Social Science) win 18.0 프로그램을 활용하여 분석하였으며, 교육 프로그램 실시 사전·사후의 논리적 사고력과 컴퓨팅 사고기반의 문제해결능력을 파악하기 위하여 항목에 대해 대응표본 *t-test*를 실시하였다.

4.4 검사결과

4.4.1 논리적 사고력 검사결과

본 연구에 사용된 검사 도구는 Piaget 이론 기반의 표준 논리 검사 요소인 계열화논리, 비례논리, 조합논리, 명제논리, 변인통제논리, 확률논리 등 총 6가지 요소에 대해 측정할 수 있는 논리적 사고력 검사지로 사전검사 A형, 사후검사에 동형 검사지인 사후검사 B형을 투입하였다[25]. 사용한 검사지는 선다형 검사로 논리적 사고에 관련한 6개의 하위변인에 따라 각 2문항씩 총 12문항으로 구성되어 있다. 중학생의 논리적 사고력 측정을 위한 논리적 사고 검사지 A, B형의 각 하위요인별 구성은 <표 11>과 같다.

<표 11> 논리적 사고력 검사지 사전·사후의 하위 문항 수

하위요소별 측정요인	문항수	문항번호	비고
계열화논리	2	(1), (2)	논리적 사고 검사 도구 (사전A),(사후B)의 각 하위요인별 2문항씩(만점2점) 총 12문항으로 구성
비례논리	2	(3), (4)	
조합논리	2	(5), (6)	
명제논리	2	(7), (8)	
변인통제논리	2	(9), (10)	
확률논리	2	(11), (12)	
계	12		

이러한 하위요인별 측정요인의 평균표준점수를 바탕으로 SPSS win 18.0을 이용하여 유의수준 $p=.05$ 와 $p=.01$ 로 대응표본 *t-test*하였으며 논리적 사고 검사 각 하위요인별 검사결과는 실험집단 중학교 1,2학년 <표 12>, 실험집단 중학교 3학년 <표 13>과 같다.

<표 12> 중학교 1,2학년 논리적 사고력 사전·사후 검사결과

하위요소	실험처치	N	평균 표준점수	SD	t	p
계열화논리	사전	22	1.55	.60	-3.130	.005**
	사후	22	1.86	.35		
비례논리	사전	22	.86	.83	-3.480	.002**
	사후	22	1.45	.60		
조합논리	사전	22	1.73	.46	-2.160	.042*
	사후	22	1.91	.29		
명제논리	사전	22	1.41	.59	-2.017	.057
	사후	22	1.64	.49		
변인통제논리	사전	22	1.14	.89	-3.177	.005**
	사후	22	1.59	.59		
확률논리	사전	22	1.32	.89	-2.592	.017*
	사후	22	1.68	.57		
전체(6개영역)	사전	22	8.00	2.29	-6.727	.000**
	사후	22	10.14	1.25		

*: $p<.05$ **: $p<.01$ N:사례수

<표 13> 중학교 3학년 논리적 사고력 사전·사후 검사결과

하위 요소	실험 처치	N	평균 표준점수	SD	t	p
계열화 논리	사전	20	1.55	.60	-2.101	.049*
	사후	20	1.90	.31		
비례 논리	사전	20	.85	.88	-3.390	.003**
	사후	20	1.55	.51		
조합 논리	사전	20	1.80	.41	-1.371	.186
	사후	20	1.95	.22		
명제 논리	사전	20	1.60	.60	-2.349	.030*
	사후	20	1.90	.31		
변인 통제 논리	사전	20	1.65	.75	-2.101	.049*
	사후	20	2.00	.00		
확률 논리	사전	20	1.50	.76	-1.561	.135
	사후	20	1.75	.44		
전체 (6개 영역)	사전	20	8.95	2.33	-5.294	.000**
	사후	20	11.05	.83		

*: p<.05 **: p<.01 N:사례수

중학교 3학년 논리적 사고력 검사에서는 사전·사후 검사 결과 계열화 논리, 비례논리, 명제 논리, 변인통제논리 하위요소에서 각각 $p=.049(p<.05)$, $p=.003(p<.01)$, $p=.030(p<.05)$, $p=.049(p<.05)$ 으로 통계적으로 유의미한 차이를 보였다.

중학년 1,2학년의 경우 논리 검사 요소인 계열화논리, 비례논리, 조합논리, 명제논리, 변인통제논리, 확률논리 5가지 하위요소에서 사전·사후 검사 결과 모두 향상된 결과를 보여 주고 있으며 통계적으로도 유의미한 차이를 보여주고 있다. 또한 전 영역에서 $p=.000(p<.01)$ 으로 통계적으로 유의미한 차이를 보였다. 중학교 3학년의 경우 계열화 논리, 비례논리, 명제논리, 변인통제논리 각 하위 요소에서 유의수준 $p=.05$ 와 $p=.01$ 이하지만 조합 논리, 확률논리에서 사전검사 결과(하위요소별 만점 2점) 평균 1.80점과 1.50점에서 사후검사 결과 평균 1.95점과 1.75점으로 각각 0.15점, 0.25점씩 증가하였으며 통계적으로도 유의미한 차이를 보였다.

위의 통계적 내용을 바탕으로 논리적 사고력 검사결과를 분석해보면 개발한 교육 프로그램은 연구(실험)집단의 계열화논리, 비례논리, 조합논리, 명제논리, 변인통제논리, 확률논리 6가지 하위 요소에서 공통적으로 긍정적인 효과를 보이고 있으며 이는 본 교육 프로그램이 학습자의 논리적

사고력 향상에 긍정적인 효과가 있음을 나타내는 것이라 할 수 있다. 또한 중학교 3학년보다 중학교 1,2학년 학생들의 논리적 사고력 하위요소별로 더욱 고른 향상을 보이고 있어 내용체계 난이도만 적절하게 조정한다면 CT 기반의 프로그래밍 학습이 중학교 전 학년에도 충분히 투입되어 긍정적 효과를 미칠 수 있다고 사료된다.

4.4.2 컴퓨팅 사고기반 문제해결능력 검사결과

본 연구의 대상에 대한 ‘컴퓨팅 사고기반의 문제해결능력 문항지’ 응답결과를 바탕으로 유의수준 $p=.05$ 와 $p=.01$ 로 대응표본 $t-test$ 를 실시하였다. 사전·사후 검사결과는 아래의 <표 14>와 같다.

<표 14> 컴퓨팅 사고기반 문제해결능력 검사결과

영역	집단별 실험처치	N	평균 표준점수	SD	t	p	
총점 평균	1,2 학년	사전	22	5.27	2.35	-2.773	.011*
		사후	22	6.05	1.73		
	3학년	사전	20	5.45	1.76	-3.943	.001**
		사후	20	6.55	1.61		
	1~2 학년 (전체)	사전	42	5.36	2.07	-4.707	.000**
		사후	42	6.26	1.62		

*: p<.05 **: p<.01 N:사례수

<표 14>의 컴퓨팅 사고기반 문제해결능력 검사결과는 10문항의 총점(10점 만점)을 토대로 실험 집단별 평균을 비교한 결과이다. 위 결과를 살펴보면 중학교 1,2학년 실험집단의 경우 컴퓨팅 사고 기반의 문제해결능력 검사지에서 사전검사 결과 총점 평균 5.27점에서 사후검사 총점 평균 6.05점으로 평균 0.78점 증가하였으며 통계적으로 유의확률 $p=.011(p<.05)$ 로 유의미한 차이를 나타냈다.

중학교 3학년 실험집단의 경우는 사전검사 결과 총점 평균 5.45점에서 사후검사 총점 평균 6.55점으로 평균 1.10점 증가하였으며 유의확률 $p=.001(p<.01)$ 로 통계적으로 유의하였다. 또한 중학교 1~3학년 전체 사전·사후 검사에서는 사전 검사 결과 5.36점에서 사후검사 결과 6.26점으로

0.90점 증가하여 통계적으로 유의한 차이를 나타냈다. 중학교 1,2학년 실험집단의 경우 논리적 사고력 검사결과, 중학교 3학년 실험집단보다 더 많은 하위요소에서 유의미한 차이를 보인 반면 컴퓨팅 사고기반 문제해결능력 검사에서는 두 집단 모두 유의미한 차이를 보여주었다. 따라서 대응표본 *t-test* 결과를 토대로 분석해보면 개발한 CT 기반의 프로그래밍 학습이 논리적 사고력과 창의적 문제해결능력 향상에 긍정적인 변화를 주고 있다고 판단된다.

4.5 연구 결과 논의 및 향후 연구방향

본 연구에서 개발한 교육 프로그램을 적용 후 논리적 사고력과 컴퓨팅 사고기반 문제해결능력 검사결과 모두 유의미한 차이를 나타냈다. 이에 학생의 창의적 문제해결력 신장에 효과적이었으며, 논리적 사고의 하위요소인 계열화논리, 비례논리, 조합논리, 명제논리, 변인통제논리, 확률논리 등 모든 영역에서 긍정적인 효과를 미치는 것으로 나타났다. 또한 컴퓨팅 사고력 기반 창의력 증진을 목표로 한 교육 프로그램은 실험 집단의 컴퓨팅 사고기반의 문제해결능력 향상에 긍정적인 변화를 주고 있음을 알 수 있다. 따라서 CT 기반 프로그래밍 학습이 다양한 아이디어를 구상하고 조직하고 표현하는 복합적인 사고 과정을 통해 학생들의 창의적 문제해결력을 신장시킬 수 있는 학습의 중요한 학습내용이 될 수 있다는 점에서 그 의미가 크다. 그러므로 보다 다양하고 체계적인 교육내용과 교육 프로그램에 대한 연구가 필요하다고 사료된다. 또한 컴퓨팅 사고는 고등사고 능력이면서 동시에 다양한 요소가 융합된 복잡한 사고과정이기때문에, 향후 연구에서는 컴퓨팅 사고력에 대한 요소의 합의와 함께 그 평가 방법에 대한 구체적인 방안이 마련되어야 할 것이다.

이에 컴퓨팅 사고력의 교육이 교과 내용의 학습에 어느 정도 영향을 끼쳤는지에 대한 상관관계 연구 또한 필요할 것으로 사료된다. 아울러 초·중학교 저학년 시기부터 컴퓨팅 사고력의 필요성을 깨닫고 체험할 수 있는 기회를 부여하고 점차 학년이 올라갈수록 심화된 과정을 경험함과 동시에 다양한 교과 활동에 컴퓨팅 사고력을 활

용할 수 있도록 함으로써 실생활 속에서의 다양한 문제 해결까지 전이가 일어날 수 있도록 해야 한다.

5. 결론

CT에 대한 연구는 최근 몇 년 사이에 급격하게 진행되어왔고, 구체적인 정의 및 교육에 관련된 내용이 아직까지 정립되어 있지 않거나 미비한 실정이다. 이렇게 명확하게 규정되지 않은 내용을 바탕으로 교육 현장에서 성급하게 컴퓨팅 사고력을 도입하게 된다면 컴퓨팅 사고력의 필요성을 충분히 전달하지 못할 수 있다. 이에 컴퓨팅 사고력의 성공적인 도입을 위해 본 연구는 기존 컴퓨터 소양과 컴퓨터 활용 중심의 컴퓨터 교육에서 벗어나 학생들에게 논리적 사고력과 창의적 문제해결능력을 향상시킬 수 있는 교육내용으로 일선 현장에 적용 가능한 CT 기반의 프로그래밍 학습내용을 개발해야 한다. 이를 위해서 학교 선택교과 교육과정(2011), 초·중등학교 정보통신기술 교육 운영지침(2009), 프로그래밍 교육 및 교육용 프로그래밍 언어(EPL)관련 선행연구들을 분석하였다. 컴퓨팅 사고력 도입이 가능한 적절한 교육 영역과 학습 주제 선정 후 이를 바탕으로 컴퓨팅 사고와의 관련성, 창의적 사고와의 관련성, 컴퓨터 과학 교육 내용의 난이도, 프로그래밍 학습 적용의 용이성 등을 고려하여 중학생을 대상으로 교육 가능한 세부 교육 내용을 비롯한 이에 적절한 교수학습 전략 및 학습 도구의 선정 후 개발된 교육내용을 적용하였다. 또한 계획, 준비, 개발, 검토, 적용 및 분석의 5단계를 설정하고 이를 바탕으로 각 단계에 따라 연구를 진행하여 본 교육 프로그램을 개발 및 적용하였다. 본 연구에서 개발한 교육 프로그램은 컴퓨터를 이용한 자료와 정보의 수집 및 분석, 문제해결 절차 설계를 위한 컴퓨팅 사고 함양, 소프트웨어 기초 소양 함양 및 설계 체험 등의 내용으로 구성하였다.

CT 기반 프로그래밍 학습의 효과성 분석을 위해 논리적 사고력과 컴퓨팅 사고기반 문제해결능력 향상의 유의미한 차이가 있는지에 대한 교육적 효과 검증을 실시하였다. 소프트웨어 교육 시범학교로 선정된 W중학교 1,2학년 22명과 3학년

20명 대상으로 12주간 24차시 수업을 진행한 결과 논리적 사고력 검사와 컴퓨팅 사고기반 문제 해결능력이 사전·사후 유의미한 차이를 보였으며, 학습자의 논리적 사고력 및 창의적 문제해결능력 또한 실험 전과 비교하여 향상되었음을 알 수 있다. 결과적으로 CT 기반 프로그래밍 학습이 논리적 사고력과 창의적 문제해결능력 향상에 효과적이라 분석되는 바, 본 연구에서 개발·적용한 교육 프로그램 통해 일선 현장 가능성을 고려하여 활용 및 적용한다면 체계적인 CT 기반 컴퓨터과학 교육 및 프로그래밍 교육을 효율적으로 수행하는데 큰 도움이 될 것이다. 더 나아가 학습자들에게 고등적 사고인 논리적 사고력, 문제해결력, 창의력, 정보처리능력 등을 신장시킬 수 있을 것이다.

따라서 본 연구에서는 CT 기반 프로그래밍 학습의 현장 적용 가능성을 제시하고 그 효과성에 대해서 분석하였다. 향후 컴퓨팅 사고력을 향상시키기 위한 다양한 교육 프로그램의 개발 방향을 제시하는데 의의를 찾을 수 있다. 이에 우리나라에서 새롭게 도입할 소프트웨어 교육과정도 컴퓨터소양과 컴퓨터 활용 중심 교육이 아닌 컴퓨팅 사고력 기반의 문제의 발견 및 표현, 효율적이고 효과적인 알고리즘의 제작, 사고의 수정 등의 고등한 사고 능력 계발 중심의 컴퓨터 프로그래밍 교육 내용을 더욱 강화하여 교육할 필요가 있다. 본 연구의 대상이 중학생으로 한정되었기 때문에 추후 적용 대상을 고등학생 및 일반인으로 확대하여 교육적 효과를 검증해 볼 필요가 있다고 사료되며 교수·학습 자료 및 방법 또한 개발을 위한 지속적인 연구가 필요하다.

참 고 문 헌

- [1] 문교식(2013). Computational Thinking의 초등 교육 활용 방향. 한국콘텐츠학회. 13(6), 518-526.
- [2] 한국컴퓨터교육학회 (2014). 창의적 문제 해결 방법론. 서울. 이한미디어.
- [3] Grover, S., & Pea, R. (2013). Computational Thinking in K-12: Review of the State of the field. *Educational Researcher*, 42(1), 38-43.
- [4] 교육과학기술부 (2011). 중학교 선택 교과 교육과정. 서울. 교육과학기술부.
- [5] 교육과학기술부 (2009). 초·중등학교 정보통신 기술 교육 운영지침(개정안).
- [6] 교육과학기술부 (2012). 2009 초등개정교육과정. 서울. 교육과학기술부.
- [7] Wing, J. M. (2008). Computational Thinking and Thinking About Computing, *Philosophical Transactions of the Royal Society*, Vol. 366, No. 1881, pp3717-3725.
- [8] National Research Council of the National Academies, USA, Report of a Workshop on the Scope and Nature of Computational Thinking, 2010.
- [9] 이영준 (2014). 초중등 단계 Computational Thinking 도입을 위한 기초 연구. 한국과학창의재단.
- [10] 이좌택 (2004). 문제기반학습에 터한 로봇 제어 프로그래밍 수업이 중학생의 논리적 사고력에 미치는 효과. 박사학위 논문, 한국교원대학교 대학원.
- [11] 유중현·김종혜 (2008). 문제해결과정에서의 정보과학적 사고 능력에 대한 개념적 고찰. *정보창의교육논문지*. 2(2), 23-32.
- [12] 양창모 (2014). 교육용 프로그래밍 언어를 사용한 프로그래밍 교육의 효과에 관한 메타분석, *정보교육학회논문지*, 18(2)
- [13] 김수환 (2011). Computational Literacy교육을 통한 문제해결력 향상에 관한 연구. 박사학위 논문, 고려대학교 대학원.
- [14] 진영학·김영식 (2011). 교육용프로그래밍언어의 효과에 관한 메타분석. *컴퓨터교육학회 논문지*, 14(3), 25-36.
- [15] National Research Council of the National Academies, USA(2011), Report of a Workshop of Pedagogical Aspects of Computational Thinking, 2011.
- [16] Tucker, A., Seehorn, D., Carey, S., Moix, D., Fuschetto, B., Lee, I., O'Grady-Cuniff, D., Stephenson, C., & Verno, A. (2011). CSTA K-12 Computer Science Standards. Revised 2011. CSTA Standards Task

Force. Retrieved from http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf

- [17] 신갑천 (2010). 스크래치를 활용한 로봇 프로그래밍 학습이 논리적 사고력과 문제해결력에 미치는 효과. 석사학위 논문, 경인교육대학교 교육대학원.
- [18] 김태미 (2014). STEAM 수업 자료 개발 및 적용을 통한 중등 수학 영재의 창의성 변화 분석. 석사학위 논문, 이화여자대학교 대학원.
- [19] Inhelder, B., & Piaget, J. (1958). The early growth of logic in the child: Classification and seriation. In H. E. Gruber & J. J. Voneche (Eds.). (1977), The Essential Piaget, N.Y.: Basic Books.
- [20] 이좌택, 이상봉 (2004). 컴퓨터 프로그래밍 학습에서 논리적 사고력 측정도구의 개발과 타당화 연구. 한국컴퓨터교육학회, 7(4), 15-25.
- [21] 김종혜 (2009). 정보과학적 사고 기반의 문제 해결 능력 향상을 위한 중등 교육 프로그램. 박사학위 논문, 고려대학교 대학원.



김 경 규

2007 충북대학교
컴퓨터교육과(이학사)
2013 충북대학교 교육대학원
정보·컴퓨터교육(교육학석사)

2015~현재 충북대학교 대학원 컴퓨터과학과
박사과정

2015~현재 포항제철중학교 교사
관심분야: 컴퓨터교육(SW교육), 정보영재교육,
스마트교육, 데이터베이스시스템.

E-Mail: actboy2@hanmail.net



이 종 연

1985 충북대학교
전자계산기공학과(공학사)
1987 충북대학교 대학원
전자계산기공학과(공학석사)

1999 충북대학교 대학원 전자계산학과(이학박사)

1990~1996 현대전자산업(주) 소프트웨어연구소
현대정보기술(주) CIM사업부 책임

1999~2003 강원대학교 삼척캠퍼스 정보통신공학과
조교수

2003~현재 충북대학교 소프트웨어학과 및 컴퓨터
교육과 교수

2001~2009 IEEE member

2003~2004 한국정보처리학회 논문지편집위원
데이터베이스분과 이사 역임

2010~현재 한국컴퓨터교육학회 이사(현)

2010~현재 한국융합학회장(현)

관심분야: 질의처리 및 최적화, 데이터베이스시스템,
Biomedical Data Mining, 컴퓨터교과교육
& e-Learning, 정보학영재교육.

E-Mail: jongyun@chungbuk.ac.kr