

컴퓨팅 사고력 향상을 위한 정보소양교육에 관한 연구

김경민†

요 약

정부는 4차 산업 혁명에서 세계 경제를 주도하는 인재를 양성하기 위해 소프트웨어 소양을 갖춘 창의적 융합인재 양성을 위한 정책을 수립하고 있다. 이에 따라 대학에서는 컴퓨팅 사고력을 기반으로 한 소프트웨어 교육을 교양 과목에 채택하고 있다. 그러나 소프트웨어 교육이 학습자들에게 프로그래밍 학습에 대한 부담을 가중시켜 컴퓨팅 사고력 향상이라는 근본적인 목적을 놓칠 수 있다. 본 연구에서는 학생들에게 소프트웨어 교육의 프로그래밍 학습에 대한 부담을 줄일 수 있는 방안으로 실무에서 직접 활용 가능한 정보소양교육을 통하여 컴퓨팅 사고력을 향상시킬 수 있는 수업방식을 제시하였다. 본 연구를 통해 정보소양교육도 문제에 대한 충분한 인식, 분석 및 효율적 절차를 스스로 구성할 수 있도록 하면 컴퓨팅 사고력 향상과 아울러 기능 습득도 향상시킬 수 있음을 알 수 있었다.

주제어 : 컴퓨팅 사고력, 정보소양교육, 소프트웨어교육, 컴퓨터 활용 능력

A Study on Information Literacy Education for Enhancing Computational Thinking

Kyungmin Kim†

ABSTRACT

In the fourth industrial revolution, the government has established a policy to cultivate the creative convergent talent with software literacy in order to cultivate human resources leading the economy. Therefore, software education based on computational thinking is being adopted in liberal arts courses in universities. However it would be damaged the fundamental objectives for increasing computational thinking by a weighted stress of students to learn programming of software education. In this paper, we proposed a teaching method that can improve computing thinking through information literacy education which can be directly applied in practice as a way to reduce the burden on the programming learning of software education. Through this paper, it was found that information literacy education can improve the computing thinking as well as the computing ability by making it possible to construct enough knowledge, analysis and efficient procedures for problems.

Keywords : Computational Thinking, Information Literacy Education, SW Education, Computer Literacy

† 정 회 원: 신라대학교 컴퓨터교육학과 조교수
논문접수: 2017년 5월 2일, 심사완료: 2017년 7월 24일, 게재확정: 2017년 7월 27일

1. 서론

최근 대두되고 있는 4차 산업혁명은 정보통신 기술의 융합으로 만드는 혁명이며 이로 인하여 삶의 방식이 크게 변화될 것으로 예상된다. 정부는 4차 산업 혁명 속에서 세계 경제를 주도하는 인재를 양성하기 위해 대학에서 글로벌 경쟁력을 갖춘 소프트웨어 전문 인력과 소프트웨어 소양을 겸비한 융합인재를 양성하는 정책을 수립하고 추진하고 있다. 이에 따라, 대학들은 컴퓨팅 사고력을 핵심 역량으로 하는 소프트웨어 교육을 교양 교육 과정에 채택하고 있다[1].

컴퓨팅 사고력은 문제를 해결하기 위해 문제를 찾고 해결 방안을 만들어 이를 컴퓨터 시스템을 이용하여 해결하는 사고 과정이다. 이를 통해 학습자들의 문제 해결력을 향상시킬 수 있으며, 문제 해결에 대한 답만 찾기 보다는 해결 방법을 찾아서 이를 다른 유사한 문제에 재사용할 수 있는 능력을 기를 수 있다. 컴퓨팅 사고력은 읽기, 쓰기 등과 마찬가지로 모든 사람들이 기본적으로 갖추어야 할 사고 능력으로 기능 중심의 단순한 컴퓨터 활용 능력이나 프로그램 코딩 능력과는 다른 유형의 창의적인 문제해결 능력이다[2].

대학이 소프트웨어 교육을 교양 교육에 채택하는 궁극적인 목적은 이러한 컴퓨팅 사고력을 향상시켜 각 분야에서 일어나는 다양한 문제들을 해결할 수 있는 기초 역량을 기르기 위한 것이다 [3]. 소프트웨어 교육은 문제를 정의하고 분석하여 이를 컴퓨터 과학의 원리를 적용하여 알고리즘으로 표현할 수 있는 능력을 통해 컴퓨팅 사고력을 향상시키는데 초점을 두고 있다. 그러나 소프트웨어 교육이 사고 변화와 문제 해결 능력 향상과 같은 근본적인 목적보다 프로그래밍 언어 교육이나 소프트웨어 도구 활용 측면에 맞춰지고 있다. 이러한 교육 상황에서 학습자들은 프로그래밍 언어 문법과 프로그래밍 과정에 대한 어려움을 겪게 되어 소프트웨어 교육이 어렵다는 인식을 하게 된다[3][4][5].

최근 영국에서도 소프트웨어 교육이 너무 코딩 교육에만 강조되는 것이 문제점으로 대두되면서 컴퓨팅 사고력은 문제 해결 상황을 이해하기 위해 사용해야 한다는 연구가 진행되고 있다[6].

컴퓨팅 사고력을 향상시키기 위해 프로그램 문법이나 코딩 능력보다는 컴퓨터 시스템을 기반으로 수행되는 사회 전반의 유무형 서비스를 이해하고, 문제가 주어지면 컴퓨터 시스템을 효과적으로 이용하여 문제를 해결하는 능력이 요구된다.

본 연구에서는 이미 많은 대학에서 개설된 정보소양과목을 통해 컴퓨팅 사고력을 기를 수 있는 수업 방식을 제시하고자 한다. 정보소양과목에서는 주로 한글과 엑셀 프로그램 사용법에 대한 기능 교육을 하고 있다. 본 연구에서 제시된 수업 방식은 완성된 문서를 문제로 제시하고 스스로 작성해야할 문서를 구상해 보도록 함으로써 문서 작성에 필요한 요소들을 파악하고 각 프로그램들이 제공해 주는 자동화 기능들을 적용하여 문서를 완성할 수 있도록 하여 스스로 문제를 해결할 수 있도록 한다. 실무에서 직접 활용 가능한 정보소양교육을 통해 학습자들에게 소프트웨어 교육의 코딩에 대한 부담을 줄이면서 컴퓨팅 사고력을 향상시킬 수 있을 것으로 기대된다.

2. 컴퓨팅 사고력과 컴퓨터 관련 교양 교과목의 변화

2.1 컴퓨팅 사고력(Computational Thinking)

카네기멜론 대학의 지넷 윙 교수는 컴퓨팅 사고력을 “해결해야 할 문제를 만났을 때 컴퓨터 과학자처럼 사고하는 것”이라고 정의하고, 컴퓨터 과학의 기초적인 개념에 기반을 두고 문제 해결, 시스템 설계와 인간 행동에 대한 이해를 포함하는 개념이라고 하였다[7][8]. 즉, 사람이 문제를 접했을 때 컴퓨터보다 잘 수행 할 수 있는 것과 컴퓨터가 더 잘 수행할 있는 것을 파악하여 컴퓨터의 장점을 잘 이해하고 컴퓨터 과학자처럼 사고하여 문제를 해결하는 과정을 컴퓨팅 사고력이라고 하였다. 윙 교수는 컴퓨팅 사고력을 문제 해결 과정에서 추상화를 통해서 문제의 핵심 요소를 추출하는 능력과 이를 컴퓨팅 기기를 통해 자동화하는 능력으로 나누어 좀 더 구체적으로 정의하고 다음과 같은 특징을 제시하였다. 첫째, 컴퓨팅 사고력은 프로그래밍이 아닌 개념화에 핵심이 있다. 둘째, 모든 사람이 갖추어야 할 핵심 역량인 컴퓨팅 사고력은 단순 반복적인 기술이 아니

다. 셋째, 컴퓨팅 사고력은 인간의 사고 방법이다. 넷째, 컴퓨팅 사고력은 수학적이고 공학적 사고를 보완하며 이를 결합한다. 다섯째, 컴퓨팅 사고력은 인공물이 아닌 아이디어이다. 여섯째, 컴퓨팅 사고력은 모두를 위한 것이다[2].

교육부에서도 소프트웨어 교육 운영 지침을 발표하면서 컴퓨팅 사고력에 대해 정의하였다[9]. 컴퓨팅 사고력은 “컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력”이다. 컴퓨팅 사고력의 구성 요소는 컴퓨터로 해결 가능한 문제 형태로 구조화하기, 자료를 분석하여 논리적으로 조직하기, 모델링과 시뮬레이션 등의 추상화를 과정을 통해 자료 표현하기, 알고리즘적 사고를 활용하여 해결 방법을 자동화하기, 해결 방법을 효율적으로 수행하고 검증하기 그리고 다른 문제에 문제 해결 과정을 적용하여 일반화하기로 나누어진다.

2.2 컴퓨터 관련 교양 교과목의 변화

1980년대 이후부터 컴퓨터와 네트워크 기술의 발전에 따라 대학의 교양 교육 과정에 컴퓨터 교육이 지속적으로 확대되어 개설되었다. 대학의 컴퓨터 교육은 정보소양 측면에서 진행되어 문서작성, 프레젠테이션, 스프레드시트 등 사무용 프로그램 위주의 정보소양 교육이 교양 필수과목으로 채택되어 왔다[10][11]. 정보소양 교육은 새로운 정보 시스템, 정보 기술과 소프트웨어 도구를 이해하고 이를 능숙하게 사용하기 위한 능력이다. 2013년 유네스코 보고서에서는 정보소양 개념보다 최근의 개념으로 디지털 소양을 제시하였다. 디지털 소양은 정보사회에서 필요한 디지털 도구에 대한 지식, 기술과 태도등을 포괄하는 문제해결능력이라고 하였다. 이러한 문제해결능력은 소프트웨어 교육을 통한 컴퓨팅 사고력과 성격이 중복된다[12]. 따라서 융복합 시대에서 대학의 정보소양교육은 응용 프로그램 사용법이나 활용법 중심의 기본 소양 교육에서 벗어나 학습자들의 사고력 향상과 문제 해결력 향상을 목표로 함으로써 오늘날 활발히 논의되고 있는 컴퓨팅 사고력과 연결될 수 있다[10].

컴퓨팅 사고력을 강조하면서 소프트웨어 교육

에 대한 관심이 증가되고 있으며 대학들은 교양 교육에 점차 확대 개설하고 있다. 미래창조과학부에서 2015년에 소프트웨어(SW) 중심대학을 선정하였고 이들 대학을 중심으로 신입생들을 대상으로 소프트웨어 기초 교육을 필수 교양과목으로 선택하고 있다[1].

소프트웨어 교육은 논리적 사고와 문제 해결력을 향상시키기 위해 컴퓨터를 사용하여 학습 환경을 효과적으로 제공하며, 구조적이고 형식적인 사고로 문제를 파악하고, 체계적이고 정확한 방법으로 주어진 정보를 활용할 수 있다[13]. 이러한, 소프트웨어 교육의 궁극적 목적은 컴퓨팅 사고력을 기르는 것이다. 컴퓨팅 사고력을 위한 교육은 학습 내용을 반복적으로 구성하고 스스로 탐색하여 기능을 발견하도록 하고 동료와의 교류를 통해 컴퓨터 과학의 개념과 원리를 학습하는 활동 중심의 학습내용으로 구성되어야 한다[4].

그러나 소프트웨어 교육을 받는 학습자들은 프로그래밍 문법을 학습하기 힘들어 하며 알고리즘과 프로그래밍 전략을 이해하기 어려워한다고 알려져 있다. 소프트웨어 교육을 어려워하는 요소는 문제를 이해하는 단계에서부터 코딩하는 단계까지 모두를 어려워하고 이런 상태에서 코딩에 대한 인지적 부담이 가중된다[3][5][14]. 따라서 소프트웨어 관련 교과목이 교양 과목으로 채택되어 단지 프로그래밍 교육만을 강조한다면 소프트웨어 교육은 가르치는 사람과 배우는 사람 모두에게 어렵고 힘든 교과목의 하나가 될 것이다.

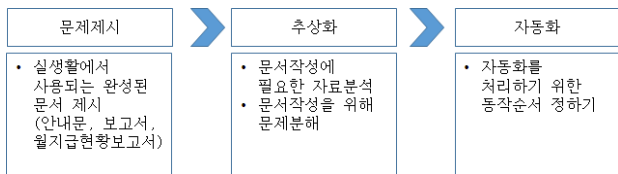
2012년 조사에 따르면 정보교육을 교양 필수과정으로 운영하는 71%의 대학에서 컴퓨터 활용 및 정보 활용능력 과목으로 엑셀 프로그램을 활용하고 있다[15]. 엑셀은 반복, 재귀적 계산을 요하는 문제나 수학적 문제를 해결하기 위한 알고리즘과 모델을 개발하는 데 용의하고 계산 결과 과정을 보여 줌으로써 전체적으로 계산 결과가 어떤 패턴으로 어떠한 변화를 가져오는 지를 쉽게 보여준다[16]. π , e 글 프로그램은 문서의 일괄성을 유지하기 위해 필요한 개요, 머리말/꼬리말, 바탕쪽과 자동으로 추출되는 차례만들기 등을 통해서 문서를 보다 체계적으로 만들 수 있다.

또한, 엑셀과 π , e 글 프로그램은 실생활 속에서 사용되는 대부분의 문서를 작성하는 프로그램

로 이러한 프로그램의 활용 능력은 실생활에서 필수적으로 요구되는 컴퓨터 소양이다. 따라서 이러한 응용 프로그램을 통해 학습자가 문제를 파악하고 스스로 문제 해결을 위한 자료 탐색과 문제 해결에 요구되는 요소를 찾아 자동화 과정을 접목시킬 수 있다면 실생활에서 필요한 소양교육과 아울러 컴퓨팅 사고력을 기를 수 있을 것이다. 소프트웨어 교육을 통한 컴퓨팅 사고력을 기르기 위한 전 단계에서 학습자들이 컴퓨터 관련 교과목이 기능 습득을 위한 교육이 아니라 실생활의 문제 상황을 정확히 인식하여 해결방안을 찾아서 제대로 동작하도록 설계하는 능력인 컴퓨팅 사고력을 기르는 교과목이라는 것을 인식하게 될 것이다.

3. 문제 중심의 정보소양교육 수업설계

기존의 정보소양교육에서는 엑셀이나 h · L 글과 같은 문서작성 프로그램의 활용법을 익히기 위해 응용 프로그램의 기능중심으로 수업을 진행하였다. 수업에 사용되는 문제 또한 기능을 익히기 위한 단편적인 예제 문제가 제시되었다.



[그림 1] 수업진행과정

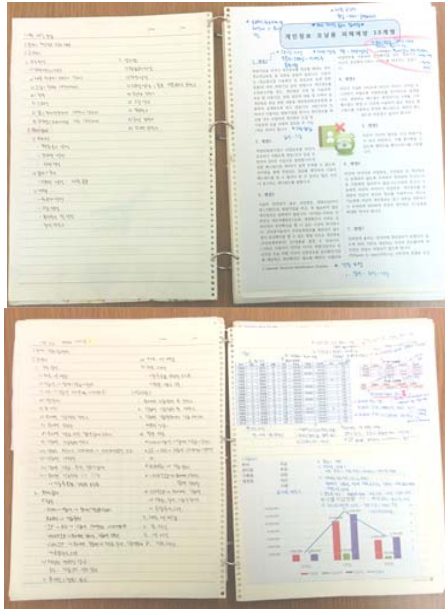
본 연구에서는 안내문, 보고서, 월지급현황보고서와 같은 실생활에서 활용될 수 있는 완성된 문서를 문제로 제시하고 문제 해결을 위한 자료수집, 문제분해, 문제해결을 위한 동작 순서 즉 알고리즘을 스스로 찾아 볼 수 있도록 하였다. 학습자는 실생활에서 사용되는 문서를 문제로 제공받음으로써 해결해야하는 문제를 쉽게 파악할 수 있었다. 또한, 주차별로 이전 주차에서 해결한 문제를 추가함으로써 반복적으로 학습할 수 있도록 내용을 구성하였다. 학습자는 반복적으로 구성된 문제를 통해서 주차가 진행되면서 좀 더 쉽게 추상화 요소를 찾아서 문제를 해결할 수 있도록 하였다.

<표 1> h · L 글과 엑셀의 추상화/자동화 요소

	추상화요소	자동화요소
h · L 글	페이지 구역나누기	구역을 나누어 구역별로 다른 형식의 문서를 작성할 수 있다.
	바탕쪽만들기	h · L 글의 다양한 개체를 이용하여 배경을 만들어 문서에 일괄적으로 배경을 적용할 수 있다.
	머리말/꼬리말적용	문서에 반복적으로 사용되는 머리말과 꼬리말을 일괄성있게 적용할 수 있다.
	개요만들기	문서 전체의 제목에 자동으로 번호를 부여하여 제목 순서가 변경되더라도 자동으로 번호가 변경될 수 있도록 할 수 있다.
	차례만들기	개요나 스타일을 이용하여 자동으로 차례를 추출해 낼 수 있다.
	다단설정하기	문서를 자동으로 여러 다단으로 나누어 작성할 수 있다.
문단	문단번호설정하기	문단에 자동으로 번호나 불릿을 설정하여 문단 변경시 번호나 불릿을 자동으로 추가 할 수 있고 번호 순서를 자동으로 변경할 수 있다.
서식	스타일 적용하기	서식을 스타일로 적용하여 서식이 변경될 때 일괄적으로 변경할 수 있다.
개체	그림/표 캡션	문서 작성에 필요한 그림이나 표를 만들어 캡션을 사용하여 자동으로 그림번호나 표번호를 입력하고 순서 변경시 자동으로 변경할 수 있다.
	주석달기	설명이 필요한 단어에 자동으로 번호를 생성하여 설명을 입력하고 원하는 위치에 배치할 수 있다.
엑셀	수식 및 함수 적용하기	수식이나 함수를 이용하여 변경된 내용을 반영하여 자동으로 계산할 수 있다.
	채우기 핸들 이용하기	채우기 핸들을 이용하여 규칙적인 내용을 자동으로 입력할 수 있다.
	데이터유효성 검사설정하기	셀에 데이터유효성 검사를 적용하여 정확한 데이터를 입력할 수 있도록 한다.
	정렬하기	입력된 자료를 원하는 순서대로 정렬할 수 있다.
	필터적용하기	입력된 자료를 원하는 자료만 선택하여 검색할 수 있도록 한다.
	차트만들기	입력된 자료를 차트로 만들어 분석할 수 있다.
	부분합만들기	입력된 자료를 특정 항목을 기준으로 자동으로 소계를 낼 수 있다.
	피벗테이블 만들기	입력된 자료를 여러 기준으로 요약, 분석, 탐색할 수 있다.

문서 작성에 필요한 추상화 요소와 자동화요소는 <표 1>과 같다. 문제를 해결하기 위해 학습자들은 문제 해결에 필요한 자료를 추출하고 분석하고 문제해결을 위한 최소 단위로 분해하는 추상화 과정과 이 과정을 통해서 분석된 데이터와 분해된 문제들을 순서를 정하고 이를 해결할 수 있는 자동화 방식을 찾는 자동화 과정을 도출할

수 있도록 하였다. 이러한 과정은 학습자들의 노트에 정리하도록 하였고 [그림 2]는 수업에 참여한 학생의 노트내용이다.



[그림 2] 학습노트

4. 연구결과

4.1 연구대상

본 연구는 부산광역시에 소재한 S대학의 2016 학년도 2학기과 동계계절학기 수업을 받은 285명의 학생을 대상으로 설문조사를 진행하였다. 교육 내용에 따른 컴퓨팅 사고력과 강의 만족도를 비교하기 위하여 박혜정, 김현숙(2014: 243-251)[17]과 최형신, 김기범(2015: 225-232)[18]의 설문지 내용을 활용하였다. 설문 문항은 컴퓨팅 사고력과 강의 만족도 영역으로 나누어 구성하였다. 컴퓨팅 사고력 영역은 컴퓨팅 사고력 구성 요소인 자료수집, 자료분석, 자료표현, 문제분해, 추상화, 알고리즘, 자동화, 시뮬레이션, 병렬화 9개의 항목[8]으로 구성하였고 강의 만족도는 기능습득, 효율적활용, 분석력, 응용력, 이해도, 집중도, 참여도, 상호작용, 참여도, 만족도등 강의를 통해 습득할 수 있는 10개의 문항으로 구성하였다. 설문문항의 내용은 <표 2>와 같다.

<표 2> 설문문항

구분	내용	
컴퓨팅 사고력	자료수집	문서의 내용을 이해하고 문서 작성을 위한 필요한 자료를 모을 수 있다.
	자료분석	수집된 자료와 문제에서 주어진 자료를 세심히 분류하고 분석할 수 있다.
	자료표현	분석된 자료 내용을 단어, 이미지, 표등으로 표현할 수 있다.
	문제분해	한글/엑셀에서 제공되는 기능으로 나누어 분석할 수 있다.
	추상화	문서 작성에서 문제의 복잡도를 줄이기 위해 한글/엑셀의 주요 기능을 사용하여 정의할 수 있다.
	알고리즘	문서 작성에 필요한 데이터 수집, 분석한 문제 해결에 필요한 요소를 순서적 단계로 나누어 표현할 수 있다.
	자동화	순서적으로 나열하고 표현한 내용을 한글/엑셀 프로그램을 이용하여 작성할 수 있다.
	시뮬레이션	문서 작성에 복잡하고 어려운 해결책을 선택하기 위해 모의 작업을 할 수 있다.
	병렬화	문서 작성시 작성된 문서를 변경하지 않고 특정 데이터만 변경하면 관련된 문서의 내용이 동시에 변경되도록 자원 구성할 수 있다.
강의 만족도	기능습득	이번 강의는 한글/엑셀의 기능을 학습하는 데 도움이 되었다.
	효율적활용	이번 강의는 학습한 기능을 효율적으로 사용하여 문서를 작성하는 데 도움이 되었다.
	분석력	이번 강의는 스스로 문서를 분석하여 문서작성을 작성하는 데 도움이 되었다.
	응용력	이번 강의는 문서작성 이외에 다른 문제를 해결하는 데 도움이 되었다.
	이해도	이번 수업방식은 수업 내용을 이해하는 도움이 되었다.
	집중도	이번 수업방식은 수업 집중도를 높이는데 도움이 되었다.
	참여도	이번 수업방식은 수업 참여도를 높이는데 도움이 되었다.
	상호작용	이번 수업방식은 교수자와 상호작용을 높이는데 도움이 되었다.
	능동성	본인은 능동적으로 수업에 참여하였다.
	만족도	수업에 만족한다.

19개의 문항은 5점 리커트 척도로 측정하였다. 설문에 대한 결과는 SPSS 23 프로그램을 이용하여 분석하였다.

<표 3> 연구 대상의 특성(N=285)

항목	빈도(명)	백분율(%)
전체	285	100%
성별	남	142 (49.82%)
	여	143 (50.18%)
소속계열	인문계열	138 (48.42%)
	자연계열	9 (3.16%)
	공학계열	102 (35.79%)
	사범계열	6 (2.11%)
	예체능계열	30 (10.53%)

응답자에 대한 일반적 특성은 <표 3>과 같다. 성별은 남자 142명(49.82%), 여자 143명(50.18%)

으로 성별의 비율은 거의 동일하였다. 응답자의 소속계열은 인문계열 138명(48.42%), 자연계열 9명(3.16%), 공학계열 102명(35.79%), 사범계열 6명(2.11%), 예체능계열 30명(10.53%)으로 인문계열과 공학계열의 학생들이 많았다.

4.2 요인분석 및 신뢰도 분석

유사 항목을 분석하기 위해 요인분석을 실시하고 유의성을 검증한 결과 KMO 값은 0.957이고, Bartlett 값이 유의 수준 p=0.000으로 요인분석을 하는 것이 적절함을 검증하였다.

추출된 요인들은 모두 0.5 이상으로 제거할 항목은 없었고 회전된 성분행렬에 의해 2개의 요인으로 그룹핑이 되는 것을 확인할 수 있었다. 신뢰도 분석결과 Cronbach 알파 값이 컴퓨팅 사고력 0.965, 강의 만족도 0.972로 매우 높게 나타나 조사도구의 적합성을 모두 충족하였고 <표 4>는 타당도와 신뢰도를 보여 준다.

<표 5>는 요인들간의 상관관계를 나타내며 모든 상관계수는 유의수준 0.01에서 0.4 이상으로

모두 통계적으로 유의미하게 정적 상관을 나타내었고 하위요인들이 서로 관련되어 있음을 알 수 있다.

<표 4> 측정도구의 타당도 및 신뢰도 분석

요인	하위요인	요인분석				신뢰도 Cronbach α
		요인 적재량	공통성	고유값	분산 설명 (%)	
컴퓨팅 사고력	자료수집	.730	.694	7.326	38.560	0.965
	자료분석	.826	.781			
	자료표현	.729	.715			
	문제분해	.844	.833			
	추상화	.802	.795			
	알고리즘	.861	.838			
	자동화	.863	.854			
	시뮬레이션	.843	.779			
강의 만족도	병렬화	.831	.790	7.841	41.271	0.972
	기능습득	.785	.768			
	효율적활용	.737	.799			
	분석력	.788	.803			
	응용력	.794	.785			
	이해도	.855	.865			
	집중도	.870	.841			
	참여도	.860	.838			
	상호작용	.859	.804			
	만족도	.796	.763			

<표 5> 하위요소 간의 상관분석

구분	컴퓨팅사고									강의만족도										
	자료수집	자료분석	자료표현	문제분해	추상화	알고리즘	자동화	시뮬레이션	병렬화	기능습득	효율적활용	분석력	응용력	이해도	집중도	참여도	상호작용	능동성	만족도	
컴퓨팅 사고	자료수집	1																		
	자료분석	.853**	1																	
	자료표현	.754**	.823**	1																
	문제분해	.713**	.769**	.736**	1															
	추상화	.682**	.707**	.670**	.824**	1														
	알고리즘	.698**	.761**	.718**	.803**	.799**	1													
	자동화	.683**	.750**	.721**	.872**	.829**	.865**	1												
	시뮬레이션	.623**	.707**	.638**	.747**	.780**	.819**	.810**	1											
	병렬화	.660**	.707**	.669**	.795**	.800**	.802**	.795**	.835**	1										
강의 만족도	기능습득	.620**	.568**	.649**	.609**	.611**	.535**	.592**	.487**	.569**	1									
	효율적활용	.683**	.662**	.726**	.675**	.663**	.637**	.658**	.603**	.611**	.844**	1								
	분석력	.627**	.609**	.650**	.607**	.628**	.605**	.623**	.583**	.565**	.810**	.865**	1							
	응용력	.582**	.574**	.625**	.583**	.617**	.574**	.594**	.565**	.585**	.758**	.791**	.832**	1						
	이해도	.604**	.559**	.610**	.602**	.617**	.606**	.612**	.568**	.565**	.788**	.806**	.802**	.815**	1					
	집중도	.535**	.514**	.585**	.542**	.570**	.531**	.548**	.495**	.549**	.750**	.732**	.760**	.777**	.844**	1				
	참여도	.549**	.519**	.566**	.569**	.636**	.559**	.575**	.518**	.528**	.737**	.729**	.753**	.770**	.836**	.862**	1			
	상호작용	.533**	.482**	.545**	.502**	.554**	.495**	.491**	.497**	.519**	.681**	.701**	.729**	.747**	.813**	.836**	.854**	1		
	만족도	.567**	.518**	.573**	.625**	.618**	.571**	.590**	.469**	.568**	.753**	.725**	.783**	.742**	.796**	.766**	.772**	.745**	1	
만족도	.563**	.520**	.566**	.561**	.559**	.542**	.541**	.443**	.519**	.764**	.742**	.733**	.741**	.839**	.822**	.839**	.822**	.761**	1	

** 상관관계가 0.01 수준에서 유의합니다(양측).

4.3 문제중심과 기능중심 방식 비교

<표 6>은 본 논문에서 제시한 컴퓨팅 사고력을 고려한 수업방식을 문제중심 방식으로 기존의 응용 프로그램의 활용 수업방식을 기능중심 방식으로 구분하였다. 수업방식의 결과를 비교하기 위하여 독립표본 t-검정을 실시하였다.

<표 6> 문제 중심 방식과 기능 중심 방식의 비교

구분	평균		표준편차		t값	p값	
	문제 중심 (N=66)	기능 중심 (N=219)	문제 중심 (N=66)	기능 중심 (N=219)			
컴퓨팅 사고	자료수집	3.924	3.658	.7084	.8495	2.555	.012*
	자료분석	3.848	3.562	.8085	.8671	2.392	.017*
	자료표현	3.939	3.680	.8015	.8716	2.155	.032*
	문제분해	3.848	3.479	.8637	.9002	2.946	.003*
	추상화	3.788	3.498	.9032	.9004	2.293	.023*
	알고리즘	3.621	3.511	.9243	.8745	.882	.378
	자동화	3.848	3.484	.8273	.9003	3.073	.003*
	시뮬레이션	3.561	3.393	.9136	.9488	1.271	.205
	병렬화	3.712	3.447	.9075	.8938	2.101	.037*
	전체	3.787	3.523	.7410	.7900	2.415	.016*
강의 만족도	기능습득	4.167	3.872	.7958	.8525	2.498	.013*
	효율적활용	3.970	3.836	.8222	.8516	1.130	.259
	분석력	3.985	3.836	.7336	.8352	1.401	.164
	응용력	3.833	3.822	.8518	.8515	.095	.924
	이해도	3.833	3.936	.9858	.8215	-.770	.443
	집중도	3.939	3.877	.9262	.8770	.502	.616
	참여도	3.848	3.913	.9486	.8762	-.516	.606
	상호작용	3.667	3.918	1.0573	.8998	-1.748	.084
	참여도	3.955	3.945	.8491	.8603	.078	.938
	만족도	3.955	3.991	.9834	.9137	-.278	.781
전체	3.9151	3.8945	.7618	.7898	.187	.851	

*p<0.05

전체 학생의 컴퓨팅 사고력과 강의 만족도를 조사한 결과 5점 만점에 컴퓨팅 사고력은 3.584점이고 강의 만족도는 3.899점으로 강의 만족도에 비해 컴퓨팅 사고력을 어려워 함을 알 수 있었다.

독립표본 t-검정 결과에서 컴퓨팅 사고력의 알고리즘 (p=0.378>0.05)과 시뮬레이션 (p=0.205>0.05) 항목을 제외하고 컴퓨팅 사고력의

모든 항목들은 p<0.05 로 교육방식에 따라 차이가 있음을 알 수 있었고 문제중심 방식이 기능중심 방식보다 대부분의 컴퓨팅 사고력 항목들이 높게 나타나 제안된 수업 방식이 컴퓨팅 사고력을 향상 시킬 수 있음을 알 수 있었다.

강의 만족도는 기능습득 (p=0.013<0.05) 항목에서만 차이를 보여 기능중심 방식(3.872점)보다 오히려 문제중심 방식(4.167점)을 통해서 기능 습득도 향상됨을 알 수 있었다.

5. 결론

본 연구는 컴퓨팅 사고력을 기르기 위해 소프트웨어 교육을 교양 교육 과정에 채택하고 있는 현 시점에서 이미 컴퓨터 교양 수업으로 개설되어 있는 정보소양교육을 통해 컴퓨팅 사고력을 향상 시킬 수 있는 수업 방식을 제안하였다. 엑셀과 등, 글 프로그램의 활용 능력은 실생활에서 필수적으로 요구되는 컴퓨터 소양 능력이다. 이러한 응용 프로그램을 활용하여 실생활에서 발생하는 다양한 문제를 학습자에게 제시함으로써 문제를 쉽게 이해할 수 있도록 하였고 문제를 해결하기 위해 필요한 자료를 수집, 분석 및 문제를 분해하는 추상화 과정과 문제 해결 순서를 정하는 자동화 과정을 학습자 스스로 수행할 수 있도록 하였다.

수업 초기 일방적으로 수업을 전달받는 주입식 수업에 익숙해져 있는 학습자들은 제시된 문제를 스스로 파악하는 단계부터 많은 어려움을 겪었다. 수업이 진행되면서 학습자들은 반복적 문제 요소에 접근해 가면서 문제를 파악하는 데 익숙해지고 유사한 문제와의 비교 등을 통하여 다양한 접근을 시도하였고 학습자 상호간 소통을 하였으며, 해결과정에 대한 시행착오등을 스스로 받아들이고 문제점을 찾으려고 하였다.

제시된 문제중심 수업방식과 기존의 기능중심 수업방식을 분석해 보니 제시된 방식이 컴퓨팅 사고력의 향상과 아울러 기능적인 습득도 향상됨을 알 수 있었다.

컴퓨팅 사고력의 향상을 위해서는 코딩 교육과 같은 프로그래밍 교육이나 컴퓨터 활용능력 배양보다는 문제에 대한 충분한 인식, 분석 및 효율적

절차를 스스로 구성하도록 하는 학습 과정이 필요하다는 것을 알 수 있었다. 소프트웨어 교육에 대한 관심과 정책 지원이 활발히 이루어지고 있는 시점에서 소프트웨어 교육에서도 코딩 위주의 교육보다는 문제를 파악하고 이를 해결하기 위해 자료분석, 문제분해와 같은 추상화 과정과 자동화 과정을 학습자 스스로 찾을 수 있도록 하는 교육 방식이 제시되어야 할 것이다. 아울러 실생활에서 필요로 하는 정보소양교육이 소프트웨어 교육과 함께 컴퓨팅 사고력을 기를 수 있도록 수업 방향을 모색해 보아야 할 것이다.

참 고 문 헌

[1] 소프트웨어 중심사회. 소프트웨어 중심대학, <http://www.software.kr/um/um02/um0207/um020701.do>

[2] Jeannette M. Wing (2006). Computational Thinking. *COMMUNICATIONS OF THE ACM*, 49(3), 33-35.

[3] 오경선 · 안성진 (2015). 프로그래밍이 어려운 이유와 컴퓨팅사고력간의 관계성 연구. **컴퓨터교육학회논문지**, 18(5), 55-62.

[4] 김수환 (2015). Computational Thinking 교육에서 나타난 컴퓨터 비전공 학습자들의 어려움 분석. **컴퓨터교육학회논문지**, 18(3), 49-57.

[5] 안상진 · 이영준 (2016). 예비교사의 프로그래밍 교수내용지식 향상을 위한 프로그래밍 교육프로그램 설계. **컴퓨터교육학회논문지**, 19(2), 1-10.

[6] 최숙영 (2016). 문제해결의 관점에서 컴퓨팅 사고력 증진을 위한 교수학습에 대한 연구. **컴퓨터교육학회논문지**, 19(1), 53-62.

[7] 이태욱 · 최현중 (2016). **정보교과교육론**. 서울:한빛아카데미.

[8] 한선관 · 류미영 (2016). **컴퓨팅 사고력을 위한 소프트웨어 교육**. 경기도:생능출판.

[9] 소프트웨어 교육 운영 지침 (2015). <http://www.pen.go.kr/www3/bbs/board.knf?oid=depart60800&wid=111>

[10] 정해용 (2014). IT 유창성과 컴퓨팅적 사고

개념을 이용한 대학 정보교양 교육에 관한 실증적 연구. **한국컴퓨터정보학회논문지**, 19(2), 263-274.

[11] 김양희 (2014). 대학 컴퓨터 실습 교양과목의 효율적 운영 방안 연구- 국가 공인자격증 ITQ 자격 취득을 위한 체육 대학 학생들의 수업 운용 중심으로. **교양교육연구**, 8(3), 323-352.

[12] 신수범 · 김철 · 박남제 · 김갑수 · 성영훈 · 정영식 (2017). 정보과 교육과정에서 컴퓨팅사고력과 연계한 디지털소양 교육과정 프레임워크 개발. **정보교육학회논문지**, 21(1), 115-126.

[13] 함성진 · 양창모 (2011). 스크래치를 이용한 초등학교 컴퓨터 교육과정 설계. **정보교육학회논문지**, 15(3), 413-423.

[14] 최현중 (2011). 대학 프로그래밍 강좌를 위한 프로그래밍 교육 프레임워크. **컴퓨터교육학회논문지**, 14(1), 69-79.

[15] 노현아 · 김자미 · 이원규 (2012). IT 유창성의 관점에서 대학 교양과정의 정보교육 내용 분석. **정보교육학회논문지**, 16(1), 41-49.

[16] 김영기 · 한희섭 (2011). 스프레드시트를 활용한 컴퓨터 프로그래밍 교육의 가능성에 관한 연구. **교육과학연구**, 42(2), 191-209.

[17] 박혜정 · 김현숙 (2014). 컴퓨터 교양 과목에 대한 학생들의 자가 평가와 성취도 분석. **한국컴퓨터정보학회논문지**, 19(2), 243-251.

[18] 최형신 · 김기범 (2015). 스크래치 프로그래밍이 예비교사에게 미치는 영향 : 컴퓨팅 사고 및 블룸의 텍사노미 활용 평가. **정보교육학회논문지**, 19(2), 225-232.



김 경 민

1993 신라대학교
전자계산학과(학사)
2000 신라대학교
컴퓨터교육과(교육학석사)

2008 신라대학교 컴퓨터공학(공학박사)
2015~현재 신라대학교 컴퓨터교육과 조교수
관심분야: 컴퓨터교육, 컴퓨팅사고력, SW교육
E-Mail: sillamin@silla.ac.kr