JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 20, No. 1, pp. 65-74, Jan. 2019

컴퓨팅 사고 향상을 위한 컴퓨팅 교육 모형 연구 -엑셀프로그램을 기반으로

이 수 진 · 이 민 정* · 박 윤 수
중앙대학교 다빈치교양대학

A Study on a Educational Model for Computational Thinking Development with Excel Program

Su-Jin Lee · Minjeong Lee* · Yoon-Soo Park
Da Vinci College of General Education, Chung-Ang University, Seoul 05974, Korea

[요 약]

ICT 기술의 발달과 함께 급변하는 현시대에는 전공에 상관없이 누구에게나 디지털 리터러시 역량을 갖추고 컴퓨팅 사고 기반의 문제해결 역량이 요구된다. 본 연구는 엑셀 프로그램이 데이터의 표현, 문제분해, 패턴인식, 추상화, 알고리즘 등 컴퓨팅 사고의 요소를 학습할 수 있으며 따라서 컴퓨팅 사고력 배양의 도구로 활용될 수 있다는 것을 확인하였다. 또한 엑셀 프로그램은 진로와 연계하여 컴퓨터 비전공자들에게 컴퓨팅 학습에 관한 당위성과 동기 부여를 할 수 있다. 이를 바탕으로 비전공자를 위한 컴퓨팅 교육에 엑셀 프로그램을 기반으로 활용할 수 있는 14주차의 교육 콘텐츠를 설계하였다. 본 연구는 엑셀 프로그램이 비전공자를 포함한 누구나 쉽게 컴퓨터 교육을 접할 수 있도록 다양한 관점과 시각을 반영한 컴퓨팅 교육용 학습 도구를 확대해 가는 데 기여할 것으로 기대한다.

[Abstract]

With the development of ICT technology in today's rapidly changing world, everyone needs digital literacy ability and problem solving ability based on computing thinking. This study confirms that Excel program can learn elements of Computational Thinking such as data representation, problem decomposition, pattern recognition, abstraction, and algorithm, and thus can be used as a tool to improve Computational thinking. In addition, the Excel program can be used in conjunction with the career path to give the computer specialists the right and motivation to learn computing. Based on this, we designed educational contents which can be utilized on the basis of Excel program for computing education for the non-majors. This study is expected to contribute to expanding learning tools for computing education that reflects various perspectives and views so that anyone who is to learn computing education can easily access the Excel program.

색인어 : 컴퓨팅 사고, 컴퓨터 비전공자, 엑셀 프로그램, 논리적 사고

Key word : Computational Thinking, Computer Non-majors, Excel, Logical Thinking

http://dx.doi.org/10.9728/dcs.2019.20.1.65

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 18 December 2018; **Revised** 26 December 2018
Accepted 20 January 2019

***Corresponding Author; Minjeong Lee**

Tel: +82-2-820-6926
E-mail: minjeonglee@cau.ac.kr

Ⅰ. 서 론

컴퓨터 비전공자들도 정보통신기술(ICT)의 기반이 되고 있는 동시대의 흐름을 이해하고 동시대가 요구하는 핵심 역량을 키워야 한다. 동시대가 요구하는 핵심역량은 디지털 역량(리터러시)에 있다[1]. 최숙영[1]은 전통적인 리터러시는 정보를 읽고, 쓰고, 말하는 것을 의미하지만 최근에는 미디어나 도구들을 사용할 수 있고 이를 이용하여 정보를 찾아 다양한 방법으로 활용하고 적용하는 능력이라고 말한다. 시대에 따라 리터러시 개념이 달라지고 있다. 표현하고 소통하는 방법에 있어 현재는 기술이 그 역할을 주도적으로 하고 있고 따라서 디지털 리터러시 역량을 갖는 것이 중요하다. 다양한 분야에서 컴퓨팅 기술이 중요한 역할을 하고 있고 우리가 '컴퓨팅 사고(Computational Thinking)'를 현재와 미래의 길잡이로 해답을 찾으려는 이유이다. 국내 뿐 아니라 전 세계는 2006년 지넷 윙(Jeannette Wing)[2]이 컴퓨팅 사고의 개념을 정립한 이후 컴퓨팅 사고 증진 교육에 심혈을 기울이고 있다.

컴퓨팅 교육의 목적은 프로그래밍 언어 또는 응용 프로그램을 이용하여 컴퓨터 비전공자들이 디지털 리터러시와 컴퓨팅 사고를 갖추고 문제를 해결할 수 있도록 논리력을 증진하는 데 있다. 이를 바탕으로 본 연구는 컴퓨터 비전공자를 대상으로 한 컴퓨팅 교육에 마이크로소프트사의 스프레드시트 프로그램인 엑셀을 이용해 문제 분석 및 해결 능력 증진을 목적으로 한다. 컴퓨터 과학자의 입장에서 컴퓨팅 사고란 논리력과 효율성을 기반으로 문제를 해결하는 것이다[2],[3]. 따라서 사고를 통한 논리력 증진이 무엇보다 중요한데 프로그래밍 언어로 컴퓨팅 사고를 증진하는 방식은 초보자에게는 어려움이 따른다. 스프레드시트 프로그램인 엑셀은 데이터 정렬, 필터링, 분석, 유효성 검사 뿐 아니라 데이터를 분석하여 압축적으로 중요한 목록을 가시화하는 응용프로그램이다. 따라서 프로그래밍 언어 수준의 코딩은 아니어도 문제해결을 위해 단계적이고 절차적 접근을 통한 데이터 모델링과 데이터 처리를 수행하는 목적을 지닌 컴퓨팅 작업 도구이다.

본 연구는 우선 통상적으로 문서저작도구나 통계도구 정도로 인식되어져 있는 엑셀 프로그램이 컴퓨팅 사고를 학습하는 데 활용될 수 있는지를 확인하기 위해 컴퓨팅 사고의 각 요소인 데이터의 표현, 문제분해, 추상화, 알고리즘에 해당하는 기능을 제공하는지 고찰하였다. 본 연구에서 제안한 엑셀 프로그램 기반의 컴퓨팅 교육 모델의 목표는 학습자의 컴퓨팅 사고력을 증진하여 디지털 리터러시 기반의 문제해결 역량을 배양하는 데에 있다. 이를 달성하기 위해 컴퓨팅 사고 증진을 위한 컴퓨팅 교육의 주제를 엑셀 프로그램의 기능과 문제해결 과정으로 구성한 한 학기 분량의 교육 콘텐츠를 설계하였다. 그 내용은 엑셀 프로그램을 기능의 측면보다는 기능을 수행하는 과정과 원리를 설명하는데 초점을 두고 실습을 통해 컴퓨터가 작업을 수행하는 과정을 직접 이해하도록 구성한다. 이를 통해 프로그래밍을 어려워하고 자발적 참여의지가 부족한 비전공자들에게 본인의 진로와 연계하여 필요성이 더 높고 쉽게 접근할 수 있는 엑셀 프로그램을 사용하여 컴퓨팅 학습에 대한 관심과 이해를 제고하고 이를 바탕으로 한 학습을 통해 실제 컴퓨팅 사고 기반의 문제해결 역량을 배양할 수 있을 것이다.

전 세계가 소프트웨어 교육에 집중하고 있다. 그것이 의미하는 바는 소프트웨어 교육이 IT강국이라고 명명하는 대한민국의 단순한 교육열이 아니라, 동시대가 요구하는 컴퓨팅 사고 역량 강화라는 목표점을 제시한다는 것을 알 수 있다. 그리고 디지털 리터러시로 표현하는 능력과 지넷 윙[2]이 주장했던 전공분야와 상관없이 컴퓨팅 사고를 보편화하여 문제를 해결하는 능력을 의미한다[4]. 소프트웨어 교육은 컴퓨터와의 소통을 기반으로 하므로 암기 및 반복학습 등 기존의 교육 방식을 적용하기 어렵다. 현 시대의 컴퓨팅 사고 역량 강화 교육을 위해 컴퓨터 비전공자들의 이해도와 흥미를 높일 수 있는 교육방식이 필요하다. 이에 우리는 본 연구의 목표 지점을 1회성 단발로 끝나는 교육체계가 아니라 본 교육 콘텐츠를 단계별로 발전 및 심화해갈 수 있는 초석을 다지는 것으로 한다.

Ⅱ. 연구 배경

2-1 국내외 소프트웨어 교육과 컴퓨팅 사고력

산업화에 앞장섰던 영국은 정보기술(IT)의 중요성을 인지하고 2008년 정보통신기술(ICT)교육 강화에 나선다. 그러나 영국 교육부는 컴퓨터 과학에 기반을 둔 컴퓨팅 교육으로 방향을 전환한다. 학생들 스스로가 컴퓨팅으로 문제를 해결할 수 있도록 교과목을 개편한 것이다[5],[6]. 영국의 컴퓨팅 교육과정에서 컴퓨팅 사고력의 구성 요소는 크게 '추상화'와 '자동화'로 분류된다. 우선 추상화 과정을 통해 현실세계의 문제를 모델링하고 분해한 후 공통된 패턴을 찾아 분류하거나 일반화하는 과정을 학습한다. 자동화 과정은 프로그래밍 훈련을 통해 습득될 수 있다고 보고 컴퓨터 과학의 여러 개념을 도입하여 문제해결을 위한 자동화 매커니즘을 완성한 후 실제 프로그램을 작성하고 디버깅, 테스트하도록 교육과정을 구성하였다[5].

미국의 컴퓨터과학교사협회 (CSTA: Computer Science Teachers Association)는 2011년에 'K-12 컴퓨터 과학 표준'을 발표하였는데 '컴퓨팅 사고력', '협업', '컴퓨팅 연습과 프로그래밍' 등 소프트웨어 기반의 문제해결 역량을 강조하고 있다[6]. 이 중 컴퓨팅 사고력 영역에서는 '문제를 해결하기 위한 절차', '정보의 표현 방식', '문제의 분해와 구조화', '프로그래밍', '수학, 과학 예술분야에서 모델링'과 '시뮬레이션을 통한 컴퓨팅 요소의 발견' 등의 과정을 전 학습 단계에서 다룬다.

우리나라 교육부는 2015 개정 교육과정에서 초·중등 교육에서의 소프트웨어 교육을 강조하였다[7]. 개정된 교육과정에서는 소프트웨어 교육의 목표를 컴퓨팅 사고력의 신장으로 두고 컴퓨팅 사고력의 구성요소로서 '자료 수집', '자료 분석', '구조

화', '분해', '모델링', '알고리즘', '코딩', '시뮬레이션', '일반화'의 9개 항목으로 분류하였다. 그런데 프로그래밍을 증진하는 학습 과정에는 컴퓨팅 사고력의 여러 요소가 복합적으로 반영되는 것이 일반적이다. 9개 항목의 의미와 요소 간 경계가 명확하지 않아 교육내용 설계 시 개별적, 단계적으로 학습하도록 목표를 설정하고 이를 달성하기 위해 과목을 설계하는 것에 어려움이 야기된다. 이에 따라 한국교육개발원은 정책 연구를 통해 컴퓨팅 사고력 중심의 소프트웨어 학습 모델로써 분해(D), 패턴인식(P), 추상화(A), 알고리즘(A)의 4가지 단계와 함께 프로그래밍(P) 항목이 선택적으로 포함되도록 정의한 DPAA(P) 모델을 고안하였다[8]. DPAA(P) 모델은 영국의 학습단계(key stage) 3에서 정의한 컴퓨팅 사고력의 구성 요소와도 일맥상통한다[9].

표 1. DPAA(P)모델[9]
Table 1. DPAA(P) Model[9]

Component	Content
Decomposition (D)	Breaking down a process into a set of smaller sub-processes to allow us to describe
Pattern Recognition (P)	Noticing or identifying similarities or common differences that will help us make predictions orlead us to shortcuts
Abstraction (A)	Making simple to information that is irrelevant in that context to solve a problem
Algorithm (A)	Developing a step-by-step strategy for solving a problem with Core Principles
Programming (P)	Implementation to a Subject by Computer Programming Language

2-2 프로그래밍 학습의 어려움

프로그래밍 학습에는 컴퓨터에서 다뤄지는 데이터의 유형과 데이터 처리에 대한 이해, 문제의 분해와 알고리즘 도출, 코딩 과정이 포함된다. 이 과정을 원활하게 진행하려면 컴퓨터의 구조, 동작, 특징을 이해하고 알고리즘을 컴퓨터로 구현하기 위해 프로그래밍 언어를 사용할 수 있어야 한다. 따라서 프로그래밍을 시작하는 초보 개발자에게는 접근이 쉽지 않다.

Jenkins[10]는 프로그래밍의 어려움을 문법(Syntax)과 의미(Semantic)로 구성된 다양한 기술, 알고리즘을 세분화하고 코딩하는 다양한 과정, 프로그래밍 언어의 학습, 새로운 과목에 대한 부담감, 학생들의 저조한 관심, 어렵다는 평판과 이미지, 학습시간의 조절의 어려움 등 7가지로 파악하였다. 최정원과 이영준[11],[12]은 컴퓨팅 사고력의 요소 중 순차, 반복, 조건, 이벤트를 학습하는 과정에서 나타난 초보 학습자들의 오류를 분석하였다. 김수환[13]은 컴퓨터 비전공 대학생을 대상으로 프로그래밍 학습의 어려움 분석에서 스크래치와 같이 직관적으로 이해할 수 있는 비주얼 프로그래밍 언어의 경우에도 변수, 리스트와 같은 데이터의 정의, 명령어의 사용과 응용에 있어서 어려움을 겪고 있음을 밝혔다. 고광일[14]은 컴퓨터 전공 학생들조차 초보 단계에서는 프로그래밍 학습에 어려움을 느끼기 때문에 교육용 프로그래밍 언어를 이용하여 기본적인 개념을 익힌 후 C 언어를 학습하는 수업 모형을 제안하였고, 이민정[15]은 프로그래밍 학습에 어려움을 겪는 초보 개발자를 위해 순서도를 활용하여 절차적 동작을 기술하는 훈련을 한 후 본격적 프로그래밍으로 진입하는 교육모델을 제안하였다.

2-4 엑셀 프로그램을 활용한 융합 교육

비전공자가 프로그래밍 학습에 어려움을 겪는 반면 IT 응용 프로그램의 활용 기술을 익히는 것에 대해서는 진로와 관계하여 상대적으로 내적 동기가 크다. 특히 엑셀 프로그램은 다량의 데이터를 무제한적으로 다룰 수 있는 자동화 환경과 수학의 성질을 찾고 탐구할 수 있는 학습 환경을 제공하는 장점을 제공한다. 이광상[16]은 엑셀 프로그램을 활용하여 일차함수의 그래프와 연립방정식의 해의 관계를 이해하는 데 효과가 있으며 또한 지필환경보다 흥미를 유발한다는 것을 보였고, 장승주[17]는 자동차 스케줄링 알고리즘의 시뮬레이션 테스트를 위해 산술기능을 제공하는 엑셀 프로그램을 활용하고 그 결과를 도출하였다. 김정아 등[18]은 엑셀 프로그램이 STEAM 교육에서 수학과 과학의 핵심내용을 시뮬레이션 해보는 공학과 기술을 필요로 하는 프로그램으로 활용될 수 있으며 주제에 따라 적절하며 향후 다양한 분야에 활용될 것으로 예측하였다.

따라서 비전공자를 위한 소프트웨어 교육에서 프로그래밍 과정에 본격적으로 진입하기 전에 엑셀 프로그램과 같은 문제해결 중심의 응용 프로그램을 활용하여 주어진 문제의 분해, 패턴 도출, 데이터 처리에 필요한 논리적 조건 수립 등의 과정을 코딩 없이 수행하는 과정을 미리 거친다면 자동화를 위한 문제해결 과정에 대한 두려움을 경감시킬 수 있을 것이다.

Ⅲ. 엑셀 프로그램과 컴퓨팅 사고

3-1 데이터의 표현과 처리

엑셀 프로그램은 데이터를 효율적으로 관리하는 스프레드시트 프로그램이다. 데이터를 그 의미와 연계성에 의해 배열 혹은 구조체 유형의 자료로 표현할 수 있는 워크시트와 여러 장의 워크시트가 상호 참조할 수 있어 데이터를 구조적이고 효율적으로 다룰 수 있다. 목적에 맞게 구성한 데이터의 처리 과정을 엑셀 프로그램이 제공하는 내장함수(embedded function)를 이용하여 자동화할 수 있다. 또한 그 결과물을 히스토그램, 그래프 등으로 단순화하여 시각화 할 수 있다.

컴퓨터 프로그래밍 언어를 이해하는데 가장 중요한 요소 중 하나가 데이터의 표현과 저장을 위한 변수를 이해하는 것이다. 변수는 데이터를 입력받아 저장하고 알고리즘 수행 과정에서 참조되고 변환된다. 그러나 프로그래밍을 처음 학습할 때 대다수의 학생들은 대체로 변수명 자체와 변수가 실제 보유한 데이터를 구분하기 어려워하며 변수가 변할 수 있는 데이터 즉 값이

라는 인식을 이해하는데 시간이 소요된다. 엑셀 프로그램은 그림 1에 보이는 바와 같이 하나의 셀(Cell)에 담길 수 있는 값을 '$A1', '$B1'과 같이 셀의 행 번호와 열 이름으로 지칭하도록 되어 있어 변수와 변수명의 관계, 즉 변수명은 변수를 담고 있는 '그릇' 이름이라는 개념을 직관적으로 이해할 수 있다.

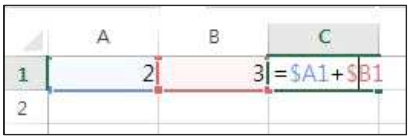


그림 1. 셀과 셀 이름을 이용한 수식의 적용
Fig. 1. Applying the cell reference into formula of a target cell

이처럼 행과 열을 이루는 셀에 데이터를 구성하는 과정은 프로그래밍에서 중요한 자료구조 형태를 반영한다. 자료구조는 자료의 특성과 크기, 주요 사용법과 수행하는 연산의 종류, 구현에 필요한 기억 공간 크기에 따라 여러 가지 종류의 자료구조 중 하나를 선택할 수 있다[19].

엑셀 프로그램에서 데이터를 프로그래밍 언어에서의 배열, 리스트, 구조체 등과 유사하게 구성할 수 있는데 이는 비전공자에게 문제를 해결하기 위해 데이터를 구조화하는 과정을 학습하는 효과가 있다. 데이터 구조를 엑셀 프로그램에서 구체화하기 위해서는 구조체, 레코드, 필드와 같은 개념이 필수이며 컴퓨터가 데이터를 다루는 방식을 직접적으로 확인할 수 있다. 엑셀 프로그램을 이용하여 문제를 해결하는 과정의 시작은 행과 열에 데이터를 할당하는 것이다. 필드에 어떤 데이터를 할당할 것인지 구조체 혹은 레코드를 어떻게 구성할 것인지 자료를 모델링하고 구조화하여 적용하는 것에서 자료 수집, 자료 분석, 구조화와 관련한 컴퓨팅 사고가 증진한다.

3-2 컴퓨팅 사고의 요소

1) 문제 분해 (Decomposition)

복잡한 문제를 해결하기 위해서는 목표한 결과를 도출하기 위한 해결 단계를 파악하고 여러 개의 작은 단위 문제로 분해를 하여 하나씩 풀어가는 방식으로 접근한다[9].

엑셀 프로그램으로 복잡한 문제를 해결할 때에도 동일한 과정을 수행한다. 즉, 입력된 데이터를 이용하여 목표한 결과를 도출하기 위해 거쳐야 할 단계와 중간 결과를 정의한 후 각 단계에서 이전 단계의 결과를 활용하도록 하는 것이다.

그림 2는 학생의 중간고사와 기말고사의 평균이 60점을 넘으면 통과(PASS)를 주고 그보다 작으면 실패(FAIL)를 결정하는 문제이다. 이 문제는 다음과 같이 3개의 과정으로 나누어 볼 수 있다.

(1) 중간고사와 기말고사의 총합을 구한다.
(2) 평균을 구하기 위해 총합을 2로 나눈다.
(3) 평균이 60보다 크면 Pass, 그렇지 않으면 Fail로 정한다.

이로부터 입력 값은 C열과 D열에 배정되는 중간고사 점수와 기말고사 점수이며 이들을 합하여 위의 (1)단계가 수행되고 그 결과가 E열에 저장된다. 다시 (2)의 평균을 구하기 위해 E2의 값을 2로 나누어 F열에 할당한다. 마지막으로 (3)의 조건에 의해 Pass 혹은 Fail이 결정된다.

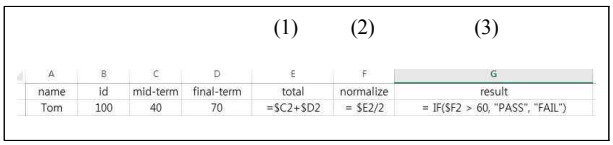


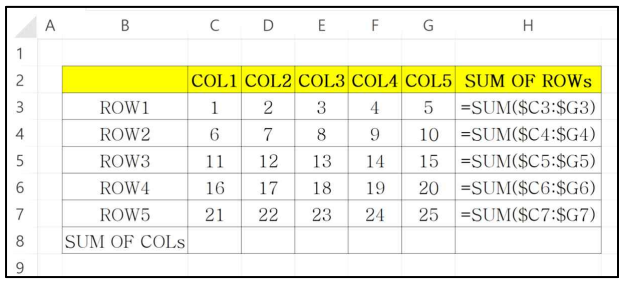
그림 2. 복잡한 문제의 분해
Fig. 2. Decomposition of a Complicated Problem

엑셀 프로그램에서 문제를 해결하기 위해 데이터의 구조화와 함께 데이터를 처리는 복잡한 문제를 단계별로 처리하는 과정을 직접적으로 구성함으로써 컴퓨팅 사고의 문제분해 과정을 경험할 수 있다.

2) 패턴 인식 (Pattern Recognition)

컴퓨팅 사고는 문제를 정형화하여 단계별로 해결하도록 사고하는 과정이다. 이 때 중요한 것은 정형화 해법을 가장 효율적인 방법으로 끌어내는 것인데 이 과정을 반복적으로 활용할 수 있도록 패턴화하는 것을 포함한다[20].

엑셀 프로그램은 복잡한 문제를 간단한 과정의 방식으로 해결한다는 데 강점이 있다. 패턴인식은 앞서 언급한 문제분화 과정에서와 같이 복잡한 문제를 단계별로 분해한 후 각 기능을 효율적으로 수행할 수 있도록 적용할 데이터의 영역과 처리 과정의 규칙을 발견하는 과정이다. 예를 들어 주어진 2차원 데이터의 각 행의 합을 구하는 문제는 우선 한 행의 값을 구하는 과정을 도출한 후 해당 과정을 모든 행에 적용하도록 문제를 해결한다. 이때 도출한 패턴은 각 행을 더하는 과정이며 이를 모든 행에 반복 적용하게 된다. 그림 3은 이 문제를 해결하기 위해 엑셀 프로그램의 SUM() 함수의 자동채우기 기능을 활용한 예를 보여준다. 1행에 해당하는 (C3~G4)의 합을 셀 H3에서 계산하여 저장하도록 한다. 이렇게 발견한 패턴을 5행까지 반복 적용할 수 있다.



	COL1	COL2	COL3	COL4	COL5	SUM OF ROWs
ROW1	1	2	3	4	5	=SUM($C3:$G3)
ROW2	6	7	8	9	10	=SUM($C4:$G4)
ROW3	11	12	13	14	15	=SUM($C5:$G5)
ROW4	16	17	18	19	20	=SUM($C6:$G6)
ROW5	21	22	23	24	25	=SUM($C7:$G7)
SUM OF COLs						

그림 3. SUM() 함수의 자동 채우기
Fig. 3. Fill up data automatically of SUM( ) in worksheet cells

3) 추상화 (Abstraction)

정인기[21]에 의하면 컴퓨터 과학에서 추상화는 불필요한 것을 제거하여 복잡함을 감소시키는 것이 목적이며 이를 위해 공통적인 요소를 통합하거나 여러 요소를 모아 집단화하고 불필요한 상세한 것들을 제거하는 과정으로 설명할 수 있다.

엑셀 프로그램은 하나의 파일 내에 여러 개의 워크시트를 구성할 수 있으며 각 워크시트는 서로 상호 참조할 수 있다. 이는 마치 프로그램의 모듈화를 연상하게 한다. 즉, 데이터에 대한 특정하거나 상세한 처리를 한 워크시트에서 작업하도록 한 후 다른 워크시트에서 그 결과를 활용하여 원하는 결과를 도출할 수 있다. 그림 4는 CT01 시트와 CT02 시트에서 각 시트에서 보유한 데이터의 평균과 총합을 구하는 작업을 하고 01 시트에서는 각 시트의 최종 결과값인 평균과 총합을 한눈에 알아볼 수 있도록 하는 구성을 보여 준다. 01 시트에서는 실제 각 시트의 상세한 데이터와 총합과 평균을 도출한 방법이 드러나지는 않고 그 결과만을 활용히는 것이다.

엑셀 프로그램이 프로그래밍 언어가 제공하는 함수 혹은 클래스에 의한 추상화 기능을 명시적으로 구현할 수는 없으나 이와 같이 워크시트 단위로 문제를 설계하고 구현하는 과정을 통해 컴퓨팅 사고의 추상화 개념을 충분히 훈련할 수 있다.

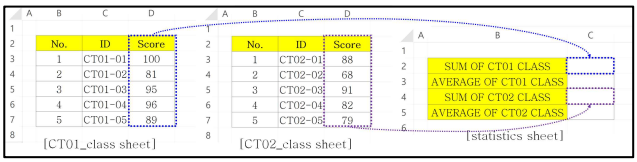


그림 4. 여러 개의 워크시트를 활용한 추상화
Fig. 4. Abstraction with multiple worksheets

4) 알고리즘 (Algorithm)

알고리즘은 데이터를 입력 받아 원하는 결과를 출력하기 위한 처리 과정의 절차를 추상화한 것이다[22]. 앞서 설명한 바와 같이 엑셀 프로그램으로 데이터의 구조화, 문제분해, 패턴매칭, 추상화 과정을 통해 복잡한 문제를 재구성할 수 있으며 이들을 해결하기 위한 단계를 절차적으로 연결하는 작업의 설계 과정을 알고리즘이라고 간주할 수 있다. 다만 통상적으로 알고리즘이 목적한 결과물을 순차적으로 도출해 내는 시간적 개념을 동반하는 동적인 활동에 관한 것이라면 엑셀 프로그램은 입력 데이터가 미리 자동화된 단계로 결과가 나오는 정적인 과정이라는 점에서 차이가 있다. 이는 마치 데이터베이스를 다루기 위한 SQL과 같은 선언형 프로그래밍 언어의 속성과 유사하다고 볼 수 있다.

최근에는 엑셀 프로그램의 데이터 혹은 각 셀의 속성이 사용자 입력에 의해 즉각적으로 반응하도록 자동화된 상호작용 기능을 제공하는 응용 소프트웨어로 그와 같은 한계를 극복할 수 있는 가능성을 보여 주고 있다. 그림 5는 엑셀 프로그램으로 마인크래프트 게임을 구현한 사례로 마인크래프트 포럼과 유투브에서 개발 방법이 공개되어 있다.

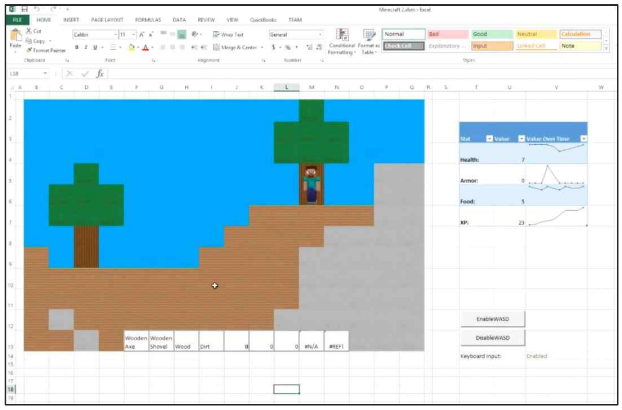


그림 5. 엑셀로 구현한 마인크래프트 게임[1)]
Fig. 5. Implementation Minecraft Game with Excel Program

Ⅳ. 엑셀 프로그램을 활용한 컴퓨팅 교육 모형

앞에서 엑셀 프로그램으로 컴퓨팅 사고력의 요소들을 증진할 수 있음을 살펴보았다. 이를 바탕으로 본 연구에서는 비전공자를 대상으로 하는 컴퓨팅 교육에 엑셀 프로그램을 활용한 14주간의 주차별 학습 내용을 표 3과 같이 설계하였다.

4-1 엑셀 프로그램을 활용한 컴퓨팅 (1주차)

엑셀 프로그램은 패턴학습을 통해 데이터를 정리하고 수식을 이용하여 계산 작업을 효율적으로 수행하는 스프레드시트 프로그램이다. 데이터를 효율적으로 파악할 수 있는 서식, 스파크라인, 테이블이 포함된 새로운 차트와 그래프를 사용하여 효과적인 방법으로 데이터를 표시한다. 클릭 한번으로 다음에 일어날 일을 예측하기도 한다[23]. 엑셀 프로그램의 기능을 잘 구현하려면 셀과 시트를 잘 활용하고 테이블과 뷰의 구성을 이해해야 한다. 다양한 내장함수를 이해하고 활용하는 것도 중요하다. 엑셀 프로그램은 다수의 함수뿐만 아니라 VBA 기능까지 내장하고 있다. 컴퓨팅 사고 증진에 있어서 엑셀 프로그램이 가지는 장점 중 하나는 수식을 작성하여 결과를 즉시 확인하고 디버깅할 수 있다는 점이다. 이는 PYTHON, MATLAB과 같은 대화형 언어의 교육과 유사하다.

컴퓨팅 사고 학습은 문제를 기술적으로 구현하는 과정으로 시작한다. 구현이 가능하도록 문제를 제대로 이해하는 것이 무엇보다 중요하며 어떤 방식으로 풀어 나가느냐 즉 알고리즘을 구현하는 것이 학습 목표이다. 알고리즘은 유한개의 규칙을 만들고 효율적으로 이행하는 절차를 만드는 것이 중요하다. 명확히 정의된 한정된 개수의 규제나 명령의 집합이기 때문이다[24]. 알고리즘의 본질은 절차를 만드는 데 있어 한정된 규칙을 적용하는 것이다.

1) https://www.minecraftforum.net/forums/, https://www.youtube.com/watch?v=gCIYqAFwr38

표 3. 엑셀을 활용한 컴퓨팅 과목의 교육내용
Table 3. The Contents of Education with Excel Program in Computational Thinking Class

week	Goals	Content	Function of Excel	CT (DPAAP)
1	To Understad Computers, SW	Operation of Computers and SW, Introduction of Computational Thinking		
2	Data Representation	Understanding Data representation within computer	Cells, Row/Column, Basic Concept of Sheets	Pattern Recognition
3	Data Input and Output	To Create a Histogram with Excel and build CSV files	CSV file, SUM, Histogram	Pattern Recognition
4	Solving Problems by Procedure	To Understand RANK() with MAX(), MIN() Production to RANK() with Weighted Sum	PRODUCT, MIN/MAX, RANK	Algorithm
5	Expression of Data	To Create Automatic Calendar in Excel	serial number, TODAY, YEAR, MONTH, DAY	Abstraction
6	Arithmetic/Comparison/Logic operations	To Create patterns with Operator and Conditional Formula	MOD, AND, OR, NOT etc	Pattern Recognition
7	A Condition and a Control	Classification of Members according to Purchase, Research to Exercise Preferences	IF, IF(AND, OR)	Decomposition Algorithm
8	To Create Complex Conditions	To Product Statistics and Grades as Scores	nested IF, COUNTIF, SUMIF, AVERAGEIF	Decomposition Algorithm
9	To Manage String Data Type	To Implement MID with Right, Left, To Understand information process with TEXT	TEXT, RIGHT, LEFT, CONCATENATE	Decomposition Algorithm
10	VBA and Loops	To Understand Loops by Flowcharts	For Loops Statement iteration	Algorithm
11	VBA and Output Statement	To Create MsgBox a Survey Program	MsgBox	Pattern Recognition
12	To Understand Random	To Create Random Numbers range to Cells with For Statement and Rnd	Rnd, Randomize, nested iteration	Algorithm Simulation
13	Assignment 1	Projects to Excel via VBA 1		Simulation
14	Assignment 2	Projects to Excel via VBA 2		Simulation

4-2 엑셀 프로그램의 자료와 입출력 (2~5주차)

엑셀 프로그램의 파일 구조는 컴퓨터의 자료 표현 방법을 이해하기 쉽게 표현하고 있다. 하나의 셀은 변수에 비유할 수 있고, 스프레드시트의 행 또는 열은 리스트에, 하나의 시트는 배열에 비유할 수 있다. 또한, 참조를 이용하여 TABLE과 VIEW의 개념에 대해 소개하는 것이 가능하다.

엑셀 프로그램에서는 동일한 자료도 적용하는 서식에 따라 다르게 표현된다. 대표적으로 일련번호(Serial Numbers)[2]를 예로 들 수 있다. 셀에 날짜를 입력하면 사용자에게는 날짜로 표시 되지만, 실제로는 셀에 일련번호가 저장된다. 일련번호는 날짜 간 덧셈과 뺄셈 연산을 수행하기 위한 것으로 학생들의 흥미를 유발할 수 있을 뿐만 아니라 컴퓨터가 자료를 어떻게 이해하고 처리하는지 소개하는 도구로 활용할 수 있다.

엑셀 프로그램을 이용하여 컴퓨터에서 일어나는 파일의 입출력을 소개할 수 있다. 예를 들어, 텍스트 편집기를 이용하여 직접 엑셀 프로그램이 읽어 들일 수 있는 형식의 파일을 만들고 이를 엑셀 프로그램에서 읽어 들여 가공한 후 히스토그램이나 그래프로 출력할 수 있다. 이와 같은 과정을 통해 다른 텍스트 프로그래밍 언어에서 사용되는 파일 입출력과 같이 컴퓨터가 자료를 받아들이고 출력하는 과정을 소개할 수 있다.

엑셀 프로그램이 데이터를 처리할 수 있는 기본 함수인 SUM(), AVERAGE(), PRODUCT(), MIN(), MAX(), RANK() 등을 학습한다. 각 함수의 기능은 단순해 보일 수 있다. 중요한 것은 쉽게 도출되는 결과도 컴퓨터는 데이터 입출력을 통해 바쁘게 계산하고 있다는 점을 이해하는 것이다. 동일한 함수도 어떻게 접근하는가에 따라서 컴퓨터의 자료표현 방법이나 컴퓨터 내부의 동작을 설명할 수 있고 실제 예제를 다루면서 절차적 문제 해결 과정을 설명할 수 있기 때문이다. 그림 6은 Score A, Score B, Score C의 점수에 정해진 가중치를 곱한 결과를 합산한 후 순위를 결정하는 문제를 해결한 결과이다. 우선 Score A, Score B, Score C의 데이터를 배열하고, 각 Score에 해당하는 가중치는 weight열에 표현하도록 설계한다. 다음 단계로 각 Score 값에 가중치를 곱한 값의 합을 구한다 (weighted_sum). 이제 문제는 weighted sum 열의 순위를 결정하는 문제로 귀결되고 이를 Rank()를 이용해 해결한다. 이처럼 한 문제를 해결하기 위한 과정을 절차적으로 설계하고 각 단계별로 해결하도록 진행한다.

	A	B	C	D	E	F	G	H	I	J
1										
2		No	Score A	Score B	Score C	weighted sum	rank		weight	
3		1	78	65	87	79.9	3		Score A	0
4		2	88	72	73	77.3	1		Score B	0
5		3	99	71	69	78.4	2		Score C	1
6		4	90	70	97	89.5	5			
7		5	61	73	96	80.9	4			
8										

그림 6. RANK()를 활용한 가중합의 정렬 문제
Fig. 6. Sorting the weighted sum of given scores with Rank()

4-3 엑셀 프로그램의 논리 연산과 조건 분기 (6~8주차)

연산자 교육에 있어서 앱인벤터와 스크래치와 같은 비주얼 프로그래밍 언어는 현실의 문제를 간략화하여 장난감 문제(toy problem)의 해결 과정을 다루는 반면 엑셀 프로그램은 일상의 상황을 조건식으로 표현하고 결론을 도출해내기에 더 효과적이다. 나아가 논리 연산자를 이용하여 복잡한 상황을 표현하는 과정에서 문제 분해, 추상화 능력을 배양할 수 있다.

일상생활에서 발생하는 문제들은 대부분 분기(Branching)를 필요로 한다. 엑셀 프로그램의 IF()는 조건식의 불린(Boolean) 값에 따라 결과를 2개로 분기하는 함수이다. 다른 프로그래밍 언어에서는 조건의 결과에 따라 일련의 동작을 수행하도록 분

2) 일련번호(Serial Numbers): 모든 날짜를 1900년 1월 1일 00시 00분 00초부터 시작되는 일련번호로 대응하여 나타낸 숫자이다.

기하는 것에 비해 엑셀 프로그램은 조건에 따라 해당 셀의 값을 바로 지정하는 것이기 때문에 단순하고 이해하기 쉽다. 이는 컴퓨팅 사고를 위한 논리 형성에 집중할 수 있는 강점이 된다. 하나의 IF()는 하나의 조건식만 판별하지만 IF()와 연산자를 조합한다면 2개 이상의 조건식을 판별하는 복잡한 문제를 설계하는 것이 가능하다. 만약 결과가 3개로 분기하거나 그 이상으로 분기해야 할 때 보통의 프로그래밍 언어와 마찬가지로 중첩 IF()를 사용할 수 있다. 그림 7은 점수(Score)의 범위에 따라 학점을 부여하는 응용문제를 해결한 것으로 학점 부여 기준을 중첩 조건문으로 작성하도록 유도한다.

	A	B	C	D
1				
2		No.	Score	Grade
3		1	78	=IF(90<=$C3,"A",IF(80<=$C3,"B","C"))
4		2	89	=IF(90<=$C4,"A",IF(80<=$C4,"B","C"))
5		3	97	=IF(90<=$C5,"A",IF(80<=$C5,"B","C"))
6		4	85	=IF(90<=$C6,"A",IF(80<=$C6,"B","C"))
7		5	72	=IF(90<=$C7,"A",IF(80<=$C7,"B","C"))
8				

그림 7. 엑셀 프로그램으로 조건분기(branching) 구현하기
Fig. 7. Implementation conditional branching

4-4 엑셀 프로그램과 알고리즘 (9~12주차)

대부분의 컴퓨팅 사고 기반의 소프트웨어 교육에서는 알고리즘 교육을 중요하게 생각하며 엑셀 프로그램은 알고리즘 교육에 적합하지 않은 툴로 인식될 수 있다. 그러나 앞서 언급한 바와 같이 엑셀 프로그램에서도 데이터 구조화, 문제분해, 패턴 매칭과 추상화 과정을 통해 컴퓨터의 알고리즘을 소개할 수 있다. 가령 엑셀 프로그램의 내장 함수인 MAX()와 MIN()를 통해 데이터의 최댓값과 최솟값을 결정하는 알고리즘을 이해할 수 있으며 이를 바탕으로 RANK()가 매개변수로 전달받은 order 값에 의해 내부적으로 어떤 알고리즘을 이용하여 순번을 부여하는지 학습할 수 있다. 예를 들어 연령 기반 데이터를 나이가 많은 순으로 정렬하는 문제를 해결하고자 할 때, MAX()와 MIN() 알고리즘을 기반으로 RANK()를 이해한 학생은 생년월일을 숫자로 바라보고 정렬 기준인 order부터 결정한 뒤 수식을 작성하도록 유도할 수 있다.

그림 8은 조건부 서식을 이용하여 패턴을 채우는 예를 보여준다. 학생들로 하여금 시트에 표현된 패턴을 보고 스스로 공통된 조건을 찾아 그 서식을 작성하도록 한다. 즉 셀의 값이 (3의 배수 + 2)인 경우 셀의 색 속성을 노란색으로 변경하는 규칙을 찾아내고 이를 실제 구현하는 방법을 안내한다. 학생들은 원하는 모양으로 패턴을 채우기 위해서는 산술적인 계산과 시뮬레이션을 통해 원하는 패턴이 나올 때까지 사고하는 과정을 경험할 수 있게 된다. 한 번 정형화 해법을 찾아낸 후에는 학생 스스로 더 복잡하고 다양한 패턴을 수식으로 표현하는 것이 가능해진다.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2	11	12	13	14	15	16	17	18	19	20
3	21	22	23	24	25	26	27	28	29	30
4	31	32	33	34	35	36	37	38	39	40
5	41	42	43	44	45	46	47	48	49	50
6										

그림 8. 조건부 서식을 이용한 패턴 채우기
Fig. 8. Fill Data automatically of SUM() in worksheet Cells

엑셀 프로그램이 주로 데이터 기반으로 데이터 변환 및 탐색 절차를 자동화하는 것을 지원하므로 다소 정적이라고 할 수 있는데 이에 대해 동적인 부분을 보완할 필요가 있다. 엑셀 프로그램에 내장된 VBA (Visual Basic for Application)로 직접 반복문은 C/C++, PYTHON의 문장 구조와 유사하고 흐름을 제어하는 문장구조를 시각적으로 확인할 수 있으므로 반복문을 교육하는 데 부족함이 없다.

	A	B	C	D	E
1	68	72	88	98	78
2	63	99	75	84	84
3	92	68	87	66	68
4	100	77	86	82	84
5	98	70	79	95	70
6					

```
Sub for_loop()
  Dim i, j As Integer
  Randomize
  For i = 1 To 5
    For j = 1 To 5
      Cells(i, j).Value = Int(Rnd() * 40) + 61
    Next j
  Next i
End Sub
```

그림 9. 엑셀/VBA로 반복문 구현하기
Fig. 9. Implementation Looping with Excel Program(VBA)

이 밖에 VBA는 Rnd()를 제공한다. Rnd()는 0에서 1사이의 실수형 난수를 생성하는 함수로, 정수형 난수를 생성하기 위해서는 컴퓨터의 내부에서 일어나는 쉬프트(shift)와 확장(scaling)에 대한 이해를 바탕으로 Int()를 이용해야 한다. 스프레드시트의 일정 영역에 원하는 범위의 난수를 생성하는 예제 설계를 통해 For 반복문과 Rnd()를 활용하여 문제 분해 능력과 알고리즘 설계 능력을 배양할 수 있다.

Ⅴ. 결 론

컴퓨터 프로그램이 일을 구현하는 방식은 설계한 절차에 따라 효율적으로 수행하는 것이다. 따라서 컴퓨팅 사고의 목표인 논리적이고 효율적으로 일을 수행하는 방법을 컴퓨팅 교육을

통해 익힐 수 있다. 그러나 컴퓨터 비전공자들에게 컴퓨팅 사고력 증진을 위한 학습으로 프로그래밍에 관심을 갖게 만드는 일은 쉽지 않다[25],[26]. 컴퓨팅 사고를 학습하는 데 난해한 요소로 동기부여(Motivation), 물리적 여건 (Physical), 기술력(Skills) 그리고 사용용도(Usage)를 들 수 있다[27]. 비전공자들이 컴퓨팅 사고의 중요성과 사고력을 키우기 위해 선택하는 프로그래밍 학습은 강력한 동기가 필요하다. 또한 프로그래밍 학습이 가능한 환경, 프로그래밍을 이해할 수 있는 수학적 지식, 마지막으로 과정을 마친 후 학생들이 유용하게 활용할 수 있어야 한다. 컴퓨팅 사고 증진의 목적은 컴퓨터 관련 전공과 무관하게 문제 해결 방식을 컴퓨팅 방식으로 해결하는 것인데 시작점에서 학생들의 무관심과 학습 당위성의 문제제기 벽을 만나게 된다. 본 연구에서는 그 시작의 높은 벽을 학습의 효율성과 기능성으로 낮추고 나아가 일상의 문제를 컴퓨팅 사고로 설계할 수 있도록 기존의 프로그래밍 언어 기반의 컴퓨팅 교육이 아닌 엑셀 프로그램을 활용하여 디지털 리터러시 역량 강화에 방향을 두었다.

엑셀 프로그램은 컴퓨터가 입력 받은 데이터를 목적에 맞게 효율적으로 구현하여 시각화하는 기능이 있다. 장점은 컴퓨터와 인간이 소통하며 일을 해결하는 방식을 직관적으로 이해할 수 있고 일상에서 기능적으로 사용할 수 있다. 단순히 문서의 저작 도구가 아니라 데이터의 구조화, 처리, 시각화를 설계해야 한다는 점에서 학생들의 문제 해석 능력, 설계 능력, 그리고 응용 능력을 키울 수 있으므로 컴퓨팅 사고 증진에 적합한 도구라고 할 수 있다. 또한, 엑셀 프로그램은 가장 많이 사용하는 범용 프로그램 중 하나로 누구나 마음만 먹으면 시작할 수 있는 프로그램이다. 비전공자들에게 컴퓨팅 사고의 당위성을 이해시키고 학습하는 동기부여를 주기에 충분하다.

본 연구에서는 엑셀 프로그램이 컴퓨팅 사고력의 각 요소인 문제 분해, 패턴 인식, 추상화, 알고리즘의 관점에서 각각의 개념을 지원하고 구현을 통해 학습할 수 있음을 살펴보았다. 또한 이를 기반으로 단순히 엑셀의 기능을 익히는 활용 중심의 교육이 아닌 일상의 문제를 해결하기 위한 과정을 절차적으로 설계하고 엑셀 프로그램을 통해 구현하는 과정을 통해 비전공자가 보다 쉽게 접근하고 받아들일 수 있는 컴퓨팅 교육 콘텐츠를 제안하였다.

소프트웨어 교육 중 컴퓨팅 사고 증진 교육의 새로운 패러다임은 교육 조건과 환경이 달라져도 적용할 수 있는 교육 방식이어야 한다. 엑셀과 같은 스프레드시트 프로그램을 이용한 학습 방식은 꾸준한 연구가 필요하다. 우리는 응용 프로그램으로 본연의 기능뿐만 아니라 컴퓨터가 프로그램을 구동하여 문제를 구현하는 과정을 이해시키고, 그 과정 중에 응용 프로그램의 기능을 논리적이고 효율적으로 활용할 수 있도록 학습과정을 설계했다. 이것은 일상의 언어로 디지털 리터러시를 구현하는 또 다른 방식이 되며 컴퓨팅 사고력 강화의 당위성이기도 하다.

향후 연구에서는 본 연구에서 제안한 엑셀 프로그램을 활용한 컴퓨팅 교육 콘텐츠를 비전공자 대상의 프로그래밍 교육에 적용한 후 학습자의 디지털 리터러시와 컴퓨팅 사고력의 증진 효과 및 계열별 특성에 따라 차이를 보이는 지를 분석할 예정이다. 이를 통해 비전공자를 위한 컴퓨팅 교육 도구로서 엑셀 프로그램이 실질적으로 활용되도록 기여하고자 한다.

참고문헌

[1] Sook Young Choi, “A Study on the Digital Competency for the Fourth Industrial Revolution,” *The Journal of Korean Association of Computer Education,* Vol. 21, No. 5, pp. 25–35, 2018.

[2] Wing, J. M., “Computational Thinking,” in *Communications of the ACM,* Vol. 49, No. 3, pp. 33–35, 2006.

[3] Lu, J. J., Fletcher and G. H. “Thinking about computational thinking,” in *ACM SIGCSE Bulletin*, Vol. 41, No. 1, pp. 260–264, 2009.

[4] Lockwood and J., Mooney, A. *Computational thinking in education: Where does it fit. A systematic literary review.* arXiv preprint, 2017.

[5] Jung Sook Sung and Hyeon Cheol Kim, “Analysis on the International Comparison of Computer Education in Schools,” *The Journal of Korean Association of Computer Education,* Vol. 18, No. 1, pp. 45–54, 2015.

[6] The Korean Academy of Science and Technology. “K-12. Computer Science Education in Korea,” in *KAST International Symposium*, 2014

[7] Ministry of Education, Software Education Guidelines, Ministry of Education: Seoul, 2015.

[8] Jin Suk Kim, Development of SW Education Teaching and Learning Model, Korea Education Development Institute, Korea Education and Research Information Service: Seoul, Technical Report CR 2015-35, 2015

[9] Seunghyun Lee and Jaehyoun Kim, “Analysis of SW basic education contents for non-majors,” in *The Proceeding of Korean Association of Computer Education*, Vol. 22, No. 2, pp. 121–124, 2018.

[10] Jenkins, T., “On The Difficulty of Learning to Program.”, in *3rd Annual LTSN-ICS Conference, Loughborough University*. pp. 53-58, 2002

[11] Jeong Won Choi and Young Jun Lee, “The analysis of Learners` difficulties in programming Learning,” *The Journal of Korean Association of Computer Education,* Vol. 17, No. 5, pp. 89–98, 2014.

[12] Kyungsun Oh and Seongjin Ahn, “A study on the relationship between difficulty in learning to program and Computational Thinking,” *The Journal of Korean Association of Computer Education,* Vol. 18, No. 5, pp. 55–62, 2015.

[13] Soo Hwan Kim, “Analysis of Non-Computer Majors` Difficulties in Computational Thinking Education,” *The Journal of Korean Association of Computer Education,* Vol. 18, No. 3, pp. 49–57, 2015.

[14] Kwangil Ko, “A Study on the EPL using Instructional Model of SW Major’s Programming Class,” *The Journal of Digital Contents Society,* Vol. 19, No. 5, pp. 891-898, May, 2018.

[15] Minjeong Lee, “Exploring the Effect of SW Programming Curriculum and Content Development Model for Non-majors College Students : focusing on Visual Representation of SW Solutions,” *The Journal of Digital Contents Society,* Vol. 18, No. 7, pp. 1313–1321, Nov. 2017.

[16] Lee Kwang Sang, “A Case Study on Application of Linear Function using Excel”, *School Mathematics*, Vol.10, No.1, pp. 1-22, 2008

[17] Seung-Ju Jang, Jae-Young Kim and Eun-Ryung Lee, “The Design of The Basic Function in The Automotive OS”, in Proceeding of the Institute of Electronics and Information Engineers, pp. 927-928, 2008

[18] Jeong-A KIM, Byeong-Su KIM, Ji-Hwon LEE and Jong-Hoon KIM, “A Study of Teaching-Learning Methods for the IT-Based STEAM Education Model With Regards to Developing People of Interdisciplinary Abilities”, *JOURNAL OF FISHRIES AND MARINE SCIENCES EDUCATION*, Vol.23, No.3, pp. 445-460, 2011

[19] Data Structure. [internet] Available: https://ko.wikipedia.org/wiki/Data_structure

[20] Aho, A. V., “Computation and computational thinking.” *The Computer Journal,* Vol. 55, No. 7, pp. 832-835, 2012

[21] InKee Jeong, “Review of Concept of Abstraction of Computational Thinking,” *Journal of The Korean Association of Computer Education*, Vol. 20, No. 6, pp. 585-596, 2016.

[22] Wing, J. M., “Computational thinking and thinking about computing,” in *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, Vol. 366, No. 1881, pp. 3717-3725, 2008.

[23] Spreadsheet Software - Microsft Excel[internet]. Available: https://products.office.com/ko-kr/excel

[24] Hyun Yule, Coding and Algorithm (with Excel VBA), *G-WORLD*, 2017

[25] Hiroyuki Aoki, JaMee Kim, Yukio Idosaka, Toshiyuki Kamada, Susumu Kanemune, and WonGyu Lee. “Development of State-Based Squeak and an Examination of Its Effect on Robot Programming Education,” in *KSII Transactions on Internet and Information Systems,* Vol. 6, No. 11, pp. 2880-2900, 2012.

[26] Su Jin Lee and Minjeong Lee, “Study of computer programming education paradigm for non-majors,” in *The Proceeding of Korean Association of Computer Education*, Vol. 21, No. 2, pp. 161–164, 2017.

[27] Kale, U., Akcaoglu, M., Cullen, T. and Goh, D. “Contextual Factors Influencing Access to Teaching Computational Thinking,” in *Computers in the Schools,* Vol. 35, No. 2, pp. 69-87, 2018.

이수진(Sujin Lee)

1996년 : 고려대학교 응용동물과학과 (농학사)
2003년 : 이화여자대학교 디자인학과 (디자인학 석사)
2007년 : 서강대학교 미디어공학과 (공학 석사)
2013년 : 서강대하교 미디어공학과 (공학 박사)

2014년~2015년: 전남대학교 강의
2015년~2017년: 교통대, 한밭대, 협성대, 중앙대, 계원예술대학교 강의
2018년~현재: 중앙대, 세종대학교 강의

※관심분야 : 인공지능기술 응용, 웨어러블 컴퓨팅 디바이스 디자인, 소프트웨어 교육 등

이민정(Minjeong Lee)

1994년 : 중앙대학교 컴퓨터공학과 (공학학사)
1996년 : KAIST 전산학과 (공학석사)

1996년~2000년: ㈜ LG전자 LG종합기술원
2000년~2010년: ㈜ 아이에이
2011년~2015년: ㈜ 삼성전자 소프트웨어센터
2016년~현 재: 중앙대학교 다빈치교양대학 조교수
※관심분야 : SW교육, 기계학습, 플랫폼, 에듀테크, AR/VR등

박윤수(Yoon-Soo Park)

2014년 : 중앙대학교 전기전자공학부 (공학학사)
2016년 : 중앙대학교 전기전자공학부 (공학석사)

2016년~현 재: 중앙대학교 대학원 전자전기공학과 (박사)
2016년~현 재: 중앙대학교 다빈치교양대학 시간강사
※관심분야 : SW교육, 정보보안, 타원 곡선 암호, 무선 통신