

엑셀을 이용한 SW 비전공자 대상 컴퓨팅 교육 모형 설계 및 효과 분석

The Analysis of Design and Effect for Software Non-majors of Computing Education Model using Excel

저자 (Authors)	박윤수, 이민정, 이수진 Yoon-Soo Park, Minjeong Lee, Su-Jin Lee
출처 (Source)	한국디지털콘텐츠학회 논문지 20(10) , 2019.10, 1969-1978(10 pages) Journal of Digital Contents Society 20(10) , 2019.10, 1969-1978(10 pages)
발행처 (Publisher)	한국디지털콘텐츠학회 Digital Contents Society
URL	http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09226297
APA Style	박윤수, 이민정, 이수진 (2019). 엑셀을 이용한 SW 비전공자 대상 컴퓨팅 교육 모형 설계 및 효과 분석. 한국디지털콘텐츠학회 논문지, 20(10), 1969-1978
이용정보 (Accessed)	중앙대학교 165.***.103.13 2020/06/25 16:10 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

엑셀을 이용한 SW 비전공자 대상 컴퓨팅 교육 모형 설계 및 효과 분석

박윤수 · 이민정 · 이수진*

중앙대학교 다빈치교양대학

The Analysis of Design and Effect for Software Non-majors of Computing Education Model using Excel

Yoon-Soo Park · Minjeong Lee · Su-Jin Lee*

Da Vinci College of General Education, Chung-Ang University, Seoul 06974, Korea

[요 약]

4차 산업 혁명은 학문 간의 경계를 없애고 정보통신기술(ICT)은 사회 전반에 걸쳐 영향을 미치고 있다. 이에 SW교육이 다양한 전공에서 컴퓨팅 기술 기반의 이해력 융합학문으로서 요구되고 있다. 이에 SW 전공자뿐만 아니라 비전공자에게도 ICT 교육의 중요성에 대한 인식이 변화하고 있다. ICT 교육의 핵심은 컴퓨팅 사고에 기반한 소프트웨어 교육에 있다. 컴퓨팅 사고 학습도구로서 스크래치, 파이썬, 랩터, 엑셀 등 다양한 프로그래밍 언어가 있고 이를 이용한 교육 모형이 있다. 본 연구에서는 기존에 제안되었던 엑셀 기반 컴퓨팅 사고 교육 모형을 정보, 패턴 인식, 알고리즘에 중점을 둔 교육으로 재편하고, SW 비전공자를 대상으로 교육을 진행하였다. 컴퓨팅 사고 역량을 측정하기 위해 학습 이전과 이후 테스트를 진행하여 통계적으로 유의미한 결과를 통해 본 연구에서 제안하는 교육 모형의 효과성을 입증하였다. 따라서 본 연구를 통해 SW 비전공자를 포함한 누구나 엑셀과 같은 ICT 활용 도구로도 컴퓨팅 교육을 효과적으로 접할 수 있다는 것을 입증했으며, 나아가 본 연구가 컴퓨팅 교육용 학습 도구를 확대해가는데 기여할 것으로 기대한다.

[Abstract]

With the advent of Industry 4.0, the discipline of Information and Communication Technology (ICT) has the potential to affect all aspects of society. Accordingly, understanding of computing technology is required as a convergence study in various majors. Therefore, the importance of ICT education is emphasized not only for SW major but also non-major. The core of ICT education is SW education based on computational thinking(CT). As a CT learning tool, there are various programming languages such as Scratch, Python, Raptor and Excel. In this study, the previously proposed Excel-based CT education model was revised into focusing on information, pattern recognition, and algorithm, and education was conducted for SW non-majors. Tests were administered before and after learning to measure CT ability and the effectiveness of the proposed education model was proved through statistically significant test results. This study showed that anyone including SW non-majors can achieve CT education effectively using Excel. It is expected that this study will contribute to expanding learning tools for computing education.

색인어 : 컴퓨터과학 비전공자, 컴퓨팅 교육, 컴퓨팅 사고, 교육모형, 엑셀**Key word** : Computer Science Non-majors, Computing Education, Computational Thinking, Educational Model, Excel<http://dx.doi.org/10.9728/dcs.2019.20.10.1969>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<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 26 August 2019; Revised 16 September 2019

Accepted 20 October 2019

*Corresponding Author; Su-Jin Lee

Tel: +82-2-886-1320

E-mail: genegraphy@sogang.ac.kr

I. 서론

4차 산업 혁명은 물리적, 생물학적, 디지털적 세계를 빅 데이터로 해석하고 활용하여 경제 및 산업 등 모든 분야에 영향을 미치는 다양한 기술로 발전한 혁명이다. 이 4차 산업 혁명이 다양한 학문 발전에 기여할 수 있었던 것은 정보통신기술 즉, ICT 기술이 뒷받침됐기 때문으로 풀이된다. 따라서 학문 간 융합에 필수인 ICT 역량의 중요성이 강조되고 더불어 ICT 교육[1]도 주목을 받고 있다. ICT 교육의 핵심은 컴퓨팅 사고(computational thinking)를 갖춘 소프트웨어 교육이다[2][3][4]. 컴퓨팅 사고는 컴퓨터와 정보에 대한 기본 개념과 원리를 바탕으로 문제를 파악하고 해결하는 일련의 사고 과정과 행동 원리를 총칭한다[5]. 다양한 학문을 기반으로 한 아카데미에서 컴퓨팅 사고를 배양하기 위한 소프트웨어 교육의 체계를 형성하기는 쉽지 않다. 대학생의 요구사항과 아카데미의 현실이 상충하는 지점이 있기 때문이다.

컴퓨팅 사고의 중요성이 교육의 일환으로 나타나기 시작한 이후 컴퓨팅 사고를 배양하는 도구에 집중해 왔다. 교육 수단인 도구의 중요성은 대학생들이 수용할 수 있는가에 대한 연구와 병행하여 변화해 왔다. 그 결과로 전공과 무관해 흥미를 갖지 못했던 소프트웨어 교육의 중요성이 강화되고 있다. 나정은[6]이 진행한 대학생들을 대상으로 한 소프트웨어 교육 필요성에 대한 설문 조사에서 “전공과 직접적 관계가 없음에도 불구하고 소프트웨어 기초교육은 필요하다”는 결론이 보고된 바 있다. 특히 “고학년 학생일수록 SW 교육의 필요성을 강하게 느끼고 있다”라는 연구 결과는 교수자와 학습자 간 소프트웨어 교육의 중요성에 대한 인식에 온도 차가 줄어들고 있음을 의미한다.

컴퓨팅 사고 교육의 목적은 컴퓨터를 이용하여 문제 분해, 패턴 인식, 알고리즘, 추상화 등 프로그래밍 요소를 훈련하고 디지털 리터러시 역량을 갖추며 마지막으로 컴퓨터를 이용하여 문제를 체계적으로 해결할 수 있는 역량을 배양하는 것이다[7]. 문제는 Papanastasiou[8]이 밝힌 바와 같이 컴퓨터의 사용이 수학적 소양과 관계가 있다는 것이다. 따라서 수학 및 정보(데이터)를 다루는 것에 초점을 두고 컴퓨팅 교육을 진행해야 한다. 엑셀을 이용하여 DPAA(P) 모델 기반 교과 과정을 설계한 이수진 외[9]의 연구를 기반으로 문제 해결 중심의 교과 과정 설계와 컴퓨팅 사고 역량 증진을 위한 문항 개발이 본 연구 핵심이다. 우리는 본 논문에서 컴퓨팅 사고 배양과 컴퓨터 프로그램의 기능을 수용할 수 있는 교육 모형을 제안하고 16주 동안 이 모형을 적용한 후 학습 전과 후, 2회의 테스트를 통해 연구의 효과성을 입증한다. 컴퓨팅 사고를 배양하는 다양한 방법 중 엑셀을 이용해 컴퓨팅 교육의 어려움을 해결하고자 하는 것이 본 논문의 목적이다.

II. 연구 배경

2-1 컴퓨팅 사고 교육 모델 분류

국내에서는 2015년에 “SW교육 교수학습 모형 개발 연구”를 통해 CT요소중심모델(DPAA(P) 모델), 시연중심모델(DMM 모델), 재구성중심모델(UMC 모델), 개발중심모델(DDD 모델), 디자인중심모델(NDIS 모델) 등 5가지로 특성화된 모델로 분류했다[10]. 우리나라 교육부는 2015 개정 교육과정에서 초·중등 교육에서의 소프트웨어 교육을 강조하였다[11]. 개정된 교육과정에서는 소프트웨어 교육의 목표를 컴퓨팅 사고력의 신장으로 두고 컴퓨팅 사고력의 구성요소로서 ‘자료 수집’, ‘자료 분석’, ‘구조화’, ‘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[10].

2-2 컴퓨팅 사고 교육 모형

분류화한 모델을 기반으로 다양한 전공생의 요구사항에 맞는 컴퓨팅 사고 교육의 도구 개발 연구가 필요하다. 따라서 컴퓨팅 사고 교육을 위한 모형이 정립되고 다양한 도구의 활용이 교육 모형으로서 효과성을 입증하고 있다. 한선관[10]은 예비교사를 대상으로 한 파이썬 기반 교육 프로그램을, 신수범[12]은 대학교 비전공자를 대상으로 한 스크래치 기반 교육 프로그램의 효과성을 검증했고 김영민, 이민정[13]은 대학교 비전공자를 대상으로 랩트 기반의 컴퓨팅 사고 수업을 설계해서 16주 과정으로 교육 프로그램의 효과성을 검증했다.

이수진 외[9]는 엑셀 기반의 교육 모형을 제시함으로써 실생활과 병행할 수 있는 컴퓨팅 교육 모형을 제시하였다. 엑셀은 스프레드시트에 저장된 자료를 계산하여 유의미한 정보를 얻어내는데 특화된 정보 처리 프로그램이다. 엑셀에서 자료를 표현하는 방법은 컴퓨터와 DB에서 자료를 표현하는 방법과 유사하여 낯설게 느껴질 수 있는 정보와 컴퓨터에 대한 지식을 다른 도구보다 쉽게 습득할 수 있는 장점이 있다. 스프레드시트를 구성하는 각 셀은 자료를 저장하는 데 사용되는 것이 아닌, 서식을 이용하여 패턴을 표현하고 이를 학습하는데 유용한 도구로 사용될 수 있어 패턴 인식 학습이 가능하다.

각각의 연구 결과는 도구에 따라 서로 다른 교육 모형을 채용하여 효과성을 검증했다. 이는 컴퓨팅 교육이 특정 프로그래밍 도구에 종속되지 않는다는 것을 의미한다. 이수진 외[9]가 제시한 엑셀 기반의 교육 모형은 프로그램의 개념을 심도 있게 학습하기에는 부족하다. 따라서 엑셀의 기능을 확장하여 VBA를 교육할 필요가 있다.

III. 엑셀과 VBA를 이용한 컴퓨팅 사고 교육

3-1 정보 중심 엑셀 기반 교육

이수진 외[9]는 프로그래밍 학습이 어려운 여러 가지 이유 중 하나로 비주얼 프로그래밍 언어라도 변수, 리스트와 같은 자료의 정의, 명령어의 사용과 응용에 어려움을 겪는 반면, 셀과 시트로 구성된 엑셀은 자료 정의가 명확하고 조작이 간편하다는 점을 들어 컴퓨팅 교육에 유리하다고 주장했다.

그림 1은 엑셀과 컴퓨터의 자료 표현 방법을 비교한 것으로, 엑셀의 장점을 표현하고 있다. 그림 1의 셀은 변수에, 행과 열은 레코드와 리스트에, 스프레드시트는 배열에, 파일은 데이터베이스에 대응된다. 이 접근법을 통해 배열 데이터를 다루기 위한 인덱스의 필요성과 사용법, 자료의 복잡성에 대해 이해하기 쉽게 교육하는 것이 가능하다. 나아가 엑셀의 ROW()와 COLUMN() 함수를 이용하여 자료의 구조를 실습해 볼 수 있는데 이는 엑셀이 이론에 그치지 않고 실습까지 제공하는 정보 교육 도구로 활용될 수 있음을 시사한다.

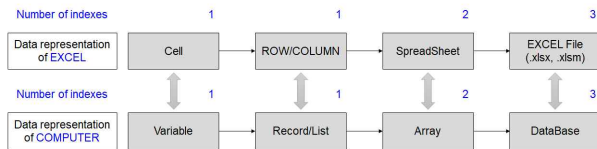


그림 1. 엑셀과 컴퓨터의 자료 표현

Fig. 1. Presentation of data by Excel and VBA

정보 처리(information processing)란 컴퓨터에서 무의미한 자료로부터 유의미한 정보를 얻어내기 위한 정보의 가공을 의미한다. 정보 처리 과정은 연산자와 제어문으로 구성하는데 엑셀은 이 연산자와 제어문을 사용하기 쉬운 함수로 제공한다. 또한 LEFT(), RIGHT(), CONCATENATE()와 같은 텍스트 함수를 제공함으로써 다양한 정보를 쉽게 가공할 수 있는 환경을 제공한다.

공공 데이터 포털[14]이나 기타 대중에게 공개된 자료는 대부분 CSV(comma separated value) 형식을 지원한다. 가령 도로 현황 자료를 이용하여 통계를 산출하고자 한다면 엑셀은 CSV 파일 형식을 지원하므로 더블 클릭만으로도 자료를 읽어 들이는 것이 가능하다. 파일을 읽어 들인 뒤에는 텍스트 함수로 자료를 추출하고, 산술, 논리 연산자와 IF() 함수로 자료를 분류하고 통계를 산출할 수 있다. 이처럼 실생활에서 정보를 가공하는데 사용되고 있는 엑셀은 함수 사용법만 학습한다면 정보 처리를 이해하는 데 효과적인 도구로 활용될 수 있다.

3-2 패턴 인식 중심 엑셀 기반 교육

엑셀의 조건부 서식은 규칙을 가진 서식 패턴을 표현하는데 적합한 기능이다. 그림 2와 같이 각 셀 안에 일정한 규칙의 숫자

를 입력하고, 특정 숫자 패턴을 의미하는 필터를 설계한 뒤 서식을 적용하면 규칙을 가진 서식 패턴이 표현된다. 필터는 조건식을 이용하여 설계하며, 산술, 논리 연산자를 이용한다면 복잡한 서식 패턴을 더 간단한 수식으로 표현하는 것이 가능하다.

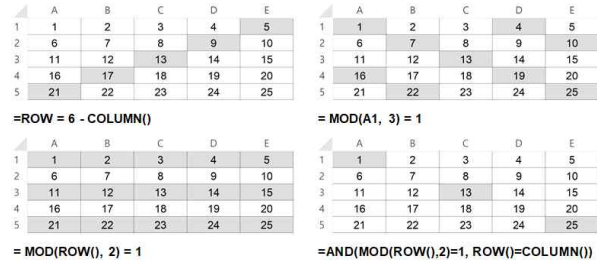


그림 2. 서식 패턴 표현

Fig. 2. Presentation of formatting data patterns

그림 3(a)는 패턴을 입력하기 위해 100개의 셀에 1~100 사이의 숫자를 입력한 것이고, 그림 3(b)는 조건부 서식으로 표현된 4개의 서식 패턴(①~④)을 적용된 순서에 맞춰 표현한 것이다. 그림 3(b)에서 조건부 서식이 중복되어 적용된 셀은 나중에 적용된 조건부 서식이 먼저 적용된 조건부 서식 위에 덮어 씌워지므로 어떤 패턴이 먼저 적용되었는지 논리적으로 추론할 필요가 있다. 다음으로 해당 패턴을 표현하기 위한 필터를 설계해야 하는데, 필터를 설계하기 위해서는 어떤 연산자를 어떻게 사용해야 할지 사고한 뒤에 수식으로 표현해야 한다. 즉, 패턴의 적용 순서를 추론하고 몇 개의 필터를 어떤 연산자를 이용해서 설계해야 하는지 판단하는 과정에서 절차와 패턴 인식 역량을 배양할 수 있다. 또한, 이를 수식으로 표현하는 과정에서 추상화 역량을 학습할 수 있을 것으로 기대해볼 수 있다.

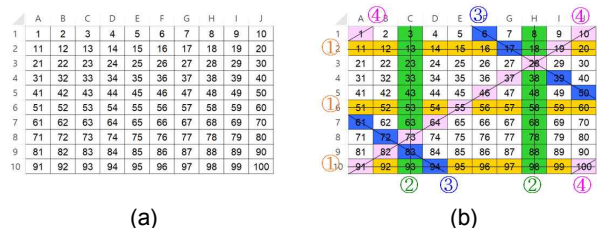


그림 3. 4개의 복잡한 서식 패턴

Fig. 3. Four patterns to describe data

3-3 엑셀과 VBA를 이용한 프로그래밍 교육

프로그램이란 특정 기능을 수행하는 데 필요한 동작을 명령어의 형태로 실행 순서에 맞춰 나열한 것이다. 엑셀에서 하나의 셀은 하나의 수식만 입력 가능하므로 명령어가 차례대로 실행되는 완전한 프로그램을 학습하기에 적합하지 않다. 또한, 엑셀은 셀에 자료를 입력할 때 자료형을 고려하지 않는다. 따라서 엑셀만 이용한다면 완전한 프로그래밍을 학습하기 어렵고 다

양한 자료형을 학습할 수 없다.

그러나 엑셀을 이용한 프로그래밍 학습에 단점만 있는 것은 아니다. 그 대표적인 도구가 논리연산자 함수다. 엑셀의 논리연산자 함수는 직관적이고 사용이 쉽다. 예를 들어 C 언어에서는 AND 연산을 위해 “조건식1 && 조건식2”의 형태로 코드를 작성해야 하므로 직관적이지 않고 수립한 조건식이 동작하기 위한 프로그램을 완성하여 실행해야 하므로 실습에 대한 부담이 크다. 반면 엑셀은 “AND(조건식1, 조건식2)”의 형태로 작성하기 때문에 논리연산자의 사용이 직관적이고 프로그램을 완전하게 작성하지 않아도 결과를 확인할 수 있다. 이는 엑셀 함수를 이용하면 논리연산자를 학습하는 데 유리하다는 것을 의미한다. 논리연산자를 능숙하게 다룰 수 있다는 것은 복잡한 조건식을 더 쉽게 표현할 수 있다는 것을 의미하는데 이는 추상화 역량을 강화하는 도구로써 엑셀이 다른 프로그래밍 도구보다 유리하다는 것을 의미한다.

엑셀을 이용한 프로그래밍 교육에서 부족한 부분은 VBA를 이용한다면 보완할 수 있다. VBA는 다른 프로그래밍 언어가 제공하는 다수의 자료형을 지원하므로 프로그래밍 기초 학습에 부족함이 없으며 완전한 프로그램을 작성할 수 있다. 표 1은 엑셀과 VBA를 이용한 프로그래밍 교육의 특징을 비교한 것으로, 엑셀과 VBA는 프로그래밍 교육에 있어 상호보완적 관계에 있음을 시사한다.

표 1. 엑셀과 VBA의 기능 비교

Table 1. Functional Comparison of Excel and VBA

	Excel	VBA
Data Type	Not supported	Provided as a grammar
Operator	Easy for learning	Difficult for learning
Branching	Functional access	Provided as a grammar
Loop	Not supported	Provided as a grammar

3-4 알고리즘 중심 VBA 기반 교육

VBA의 장점으로 VBA 코드와 스프레드시트가 자료를 주고 받을 수 있다. 이는 VBA를 이용한 프로그래밍 교육은 입력되거나 출력되는 자료를 스프레드시트를 이용하여 쉽게 확인할 수 있다는 것을 의미한다. 그림 4는 스프레드시트에 나열된 숫자를 버블 정렬을 이용하여 정렬하는 과정을 표현한 것이다. VBA로 작성한 코드의 실행 결과가 스프레드시트에 표시되므로 코드의 실행 결과를 스프레드시트에서 즉시 확인할 수 있다. 또한, 코드에 딜레이 함수를 삽입해서 코드의 실행 과정을 시각화 (visualization)할 수 있는데 이는 VBA를 이용한 프로그래밍 교육이 시각화의 장점[15]을 이용해 학습 효과를 향상할 수 있음을 시사한다.

```

Sub bubble_sort()
    Dim i, j, tmp As Integer
    For j = 1 To 9
        For i = 1 To 10 - j
            If Cells(i + 1, 1).Value < Cells(i, 1).Value Then
                tmp = Cells(i, 1).Value
                Cells(i, 1).Value = Cells(i + 1, 1).Value
                Cells(i + 1, 1).Value = tmp
            End If
        Next i
        Application.Wait (Now + TimeValue("0:00:01"))
    Next j
End Sub

```

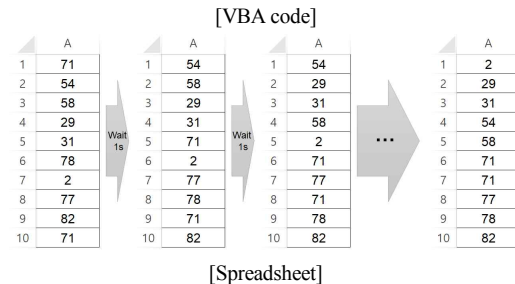


그림 4. 버블 정렬의 구현

Fig. 4. Bubble sort algorithm implementation

VBA의 Rnd() 함수는 0~1 사이의 실수형 난수를 생성하는 함수다. Int() 함수를 이용하여 정수형 난수로 변환할 수 있고 특정 값을 곱하거나 더해서 정보의 크기 조정 (scaling)과 이동 (shifting) 개념을 학습할 수 있다. 또한 난수를 생성해서 그림 5와 같이 임의의 데이터를 생성하고 제어문을 이용하여 가공하는 과정을 설계해 볼 수 있다. 그림 5와 같이 임의의 자료를 제어하기 위해서는 생성된 난수를 어떻게 제어해서 자료를 생성할지 체계적으로 설계해서 접근해야 한다. 이처럼 VBA를 이용한 난수의 학습은 프로그래밍뿐만 아니라 알고리즘을 학습하는 데 있어서 좋은 도구로 활용될 수 있다.

```

Sub Monte_carlo_method()
    Dim i, j, tmp As Integer
    For i = 1 To 10
        For j = 1 To 10
            tmp = Int(Rnd() * 10) + 1
            If tmp < 3 Then
                Cells(i, j).Value = "event A"
            ElseIf tmp < 6 Then
                Cells(i, j).Value = "event B"
            Else
                Cells(i, j).Value = "event C"
            End If
        Next j
    Next i
End Sub

```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	event B	event B	event A	event B	event B	event C	event C	event A	event C	event C		statistics	
2	event C	event B	event A	event C	event B	event C	event C	event C	event A	event B		event A	23
3	event A	event A	event B	event A	event A	event C	event C	event C	event C	event A		event B	35
4	event A	event C	event B	event B	event B	event C	event C	event C	event C	event A		event C	42
5	event B	event B	event B	event B	event B	event A	event C	event A	event A	event A		sum	100
6	event B	event B	event C	event C	event B	event C	event B	event A	event A	event A			
7	event C	event B	event B	event B	event B	event B	event A	event B	event B	event C			
8	event C	event C	event C	event B	event C	event C	event A	event B	event B	event A			
9	event C	event C	event B	event A	event B	event B	event C	event A	event A	event B			
10	event C	event C	event C	event B	event C	event C	event C	event C	event C	event B			

그림 5. 몬테카를로 방법의 적용

Fig. 5. Applications of the Monte Carlo Method

IV. 엑셀 기반 컴퓨팅 교육 모형의 설계

4-1 교육 내용

표 2. 컴퓨팅 사고 향상을 위한 엑셀 기반 컴퓨팅 교육 내용
Table 2. Contents of Excel-based computing education to improve computational thinking

week	Title	Contents	Function of Excel	CT (DPAAP)
1	To Understand Computers, SW	Operation of Computer and SW, Introduction of Computational Thinking		
2	Presentation of Data (1)	To Understand Presentation of Data in Computer	ROW/COLUMN, Index, SpreadSheet, Autofill, SUM, PRODUCT	Pattern Recognition
3	Presentation of Data (2)	To Express various Data using Excel Form, To Create Automatic Calendar	Formatting, Serial Number, YEAR, MONTH, DAY	Abstraction
4	Input Data and Output Data	To Create a Histogram with Excel and Build CSV files	CSV, Histogram, Chart	Abstraction
5	Solving Problems by Procedure	To Understand RANK() with MAX(), MIN() To Understand Flowchart	MIN/MAX, RANK, Flowchart	Algorithm
6	Arithmetic/Comparison/Logic Operations	To Understand Pattern Expression using Operator and Conditional Expression	MOD, AND, OR, NOT, Conditional Expression	Pattern Recognition
7	Expression of Pattern	To Understand Various Pattern Expressions using Conditional Formatting and Operators	Conditional Formatting	Decomposition Pattern Recognition
8	Branching and Information Processing	To Understand Branching, To Understand Information Processing using Text Functions and Control Functions	IF (AND, OR), COUNTIF, SUMIF, RIGHT, LEFT, CONCATENATE	Decomposition
9	Data Type	To Understanding the Data Management between VBA Code and Spreadsheet	Variant, Integer, Double, String, Cells	Simulation
10	Control Statement in VBA	To Understand IF Statement, To Creating MsgBox a Survey Program	MsgBox with Options, If	Decomposition Simulation
11	Loop, Increment/Decrement operator	To Understand Loops with Flowcharts, Accessing Spreadsheet Using the Loop	Cells, For Loop, Flowchart	Abstract Algorithm Simulation
12	Random	To Create Random Number, To Scaling the Range of Random Number	Randomize, If, Rnd, Int	Abstract Simulation
13	Implementation Project1	Automating the Copying Process between Spreadsheets	Data type, For Loop, If, Flowchart	Abstract Algorithm Decomposition Simulation
14	Implementation Project2	To Understand the Control of Random Data Generation	Data type, Rnd, For loop, If, Flowchart	Abstract Algorithm Decomposition Simulation
15	Algorithms with VBA	To Understand Search and Sort Algorithms	Cells, For Loop, If, Flowchart	Abstract Algorithm Decomposition Simulation

성정숙, 김현철[16]은 컴퓨팅 사고력은 그 모티브가 컴퓨터의 정보 처리 과정에서 온 것이므로 컴퓨팅 사고력을 위해서는 코딩 교육이 가장 효과적이라 기술했다. 또한, 기존 이수진 외 2[9]의 연구는 정보 처리 과정을 이해하고, VBA를 이용해서 코딩 교육을 보완하도록 개발되었으므로 컴퓨팅 사고력을 배양하기에 적합하다 할 수 있다. 이에 본 연구에서는 연산자와 조건부 서식을 이용한 추상화 및 패턴 인식 교육과, VBA를 이용한 자동화와 코딩 교육을 강화함으로써 이수진 외 2[9]가 설계한 컴퓨팅 교육 모델에서 CT 요소 교육에 집중할 수 있도록 개

편했다.

세부적으로 2주차에는 자동 채우기 (autofill)를 이용한 패턴 인식 학습과 컴퓨터의 자료 표현에 대한 교육을, 기존에 5주차에 있던 자료의 표현을 3주차에 재배치했다. 이는 2주차에 컴퓨터의 자료 표현을 이해한 뒤 3주차에 엑셀을 이용하여 직접 자료를 표현하는 방법을 교육하기 위함이다. 컴퓨터가 이해할 수 있는 파일 형식으로 자료를 입력하고, 이를 절차적으로 가공하여 히스토그램이나 그래프와 같은 시각자료로 표현하는 방법을 학습할 수 있도록 4주차 내용을 구성했다. 특히 숫자로 입력된 자료를 어떻게 히스토그램이나 그래프로 표현할 것인지 사고하고, 이를 순서대로 구현하는 과정에서 추상화와 알고리즘 역량을 강화할 수 있도록 콘텐츠를 구성했다. 5주차에는 RANK()를 이용한 절차적 문제 해결에 중점을 두고 교육할 수 있도록 내용을 구성했다. 6~7주차에 배치되어 있던 연산자, 조건과 제어는 연산자와 조건부 서식을 이용한 패턴 인식 콘텐츠로 대체하여 패턴 인식 역량 강화에 집중할 수 있도록 구성했으며, 기존의 8~9주차에 구성되어 있던 IF 조건문과 텍스트 함수의 응용은 8주차의 정보 처리의 이해로 축약했다. 대신 9주차에 자료형과 변수에 대해서 1주간 교육함으로써 체계적인 프로그래밍 교육이 가능하도록 구성했다.

10주차에는 MsgBox와 If 조건문을 이용한 응용 프로그램을 작성하는데, 어떤 설문 내용을 어떻게 가공해서 어떤 결과로 표현할 것인지 설계하는 과정에서 분해와 프로그래밍 역량을 강화할 수 있도록 콘텐츠를 구성했다. 11주차에는 반복제어(loop)와 순서도를 학습하여 알고리즘 학습의 기초를 마련한다. 12주차에는 난수를 학습하여 프로그래밍 외에도 추상화 역량을 강화할 수 있게 했다. 13주차부터는 알고리즘과 프로젝트 기반 문제 해결 과정을 강조하였다. 우선 For 반복문과 IF, 난수를 이용하여 임의의 데이터를 생성하고, 이를 다른 시트에 복사하는 과정을 자동화해보는 내용을 학습한다. 14주차에는 난수를 이용하여 임의의 자료를 생성하고 이를 제어하는 방법을 학습할 수 있도록 했다. 15주차에는 검색, 정렬 알고리즘을 직접 구현해봄으로써 알고리즘 학습 효과를 강화할 수 있도록 구성했다.

결과적으로 기존에 14주였던 교육내용을 15주로 개편했으며 정보 및 패턴 중심 교육에서는 패턴 인식 역량 강화를 알고리즘 중심 VBA 교육에서는 자동화 및 분해, 알고리즘 등 문제 해결에 초점을 맞춰 교육할 수 있도록 콘텐츠를 강화했다.

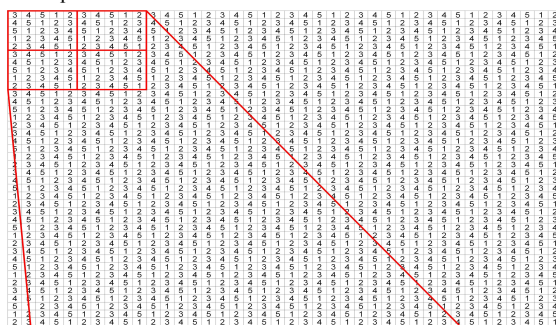
4-2 컴퓨팅 사고 역량 측정을 위한 문항 설계

컴퓨팅 사고에서 패턴 인식은 무의미한 숫자 분포에서 일정한 패턴을 찾아내고 이를 일반화하는 역량을 의미한다. 이는 본 연구에서 제안하는 ICT 교육을 통해 증진할 수 있는 역량으로, SW의 중요성이 강조되는 미래 사회에서 SW 비전공자에게도 점차 필요성이 증가하고 있는 역량이다. 패턴 인식과 마찬가지로 문제 해결 역량은 컴퓨팅 사고에서 중요한 역량이다. 최숙영 [17]은 컴퓨팅 사고는 컴퓨터를 기반으로 한 문제 해결에 관한 하나의 사고 과정으로 정의될 수 있다고 기술하고 있다. 따라서

본 연구에서는 앞서 설계한 교육의 효과성을 입증하기 위해 패턴 인식과 문제 해결 역량을 측정할 수 있는 테스트 문항을 설계했다.

1차 테스트의 첫 번째 문항은 그림 6과 같이 1~5 사이의 숫자가 규칙적으로 배치된 40×40 크기의 배열에서 제한된 시간 내에 합계를 산출하는 것으로 문제 해결 과정 또는 문제 해결을 위해 사용한 일반화된 공식을 기술하도록 설계했다. 문제 해결을 위해서는 먼저 행과 열의 수를 파악해야 한다. 40행 40열로 구성된 배열은 5행 5열마다 동일 숫자 패턴이 반복되므로, 5×5 크기의 배열이 64개 있는 것과 같고, 합계도 같은 방식으로 산출할 수 있다. 배열의 크기를 파악하고 반복되는 숫자 패턴을 파악하는 과정에서 패턴 인식 역량을, 이를 일반화된 공식으로 표현하거나 절차를 설명하는 과정에서 분해와 추상화 역량을 파악할 수 있다. 또한, 40×40 크기 배열의 합계를 산출하는 것에서 끝나지 않고 배열의 크기를 41×42까지 확장했을 때 합계와 합계를 산출한 방법과 절차를 기술하도록 장치함으로써 예외 처리를 위한 문제 해결 역량을 평가할 수 있다.

[pro1] As shown in the following figure, the number of 1 to 5 in the 1600 cells is described as a pattern, and the filled number pattern has a certain rule.



3	4	5	1	2
4	5	1	2	3
5	1	2	3	4
1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4
1	2	3	4	5
2	3	4	5	1

- * Calculate the sum of the entire sheet.
- * Describe the procedure that calculated the sum or the rule that you found while calculating the sum.

그림 6. 1차 테스트 문항 : 패턴 인식
Fig. 6. 1st test problem : pattern recognition

1차 테스트의 두 번째 문항은 그림 7과 같이 정해진 이동 규칙을 이해하여 답을 찾아내는 문항이다. 예를 들어 그림 7에서 5행 2열의 Num_2 가 59라면 9로 나눈 나머지가 5, 6으로 나눈 나머지가 5이므로 5행 5열로 이동할 수 있다. 마찬가지로 Num_2 가 40이라면 9와 6으로 나눈 나머지가 모두 4이므로 4행 4열로 이동할 수 있다. 답을 찾아내는 과정에서 9와 6의 공

배수에 1~5 사이의 숫자를 더한 값이 Num_2 가 될 수 있는 숫자라는 결론을 도출하고 이를 절차적으로 기술해야 하는 문제다. 규칙을 이해하고 일반화된 공식을 도출하는 과정에서 추상화와 알고리즘 역량을 측정할 수 있도록 설계했다.

[Pro2] Find four integers that fit into the cell that satisfy the following moving rule:

	C1	C2	C3	C4	C5
R1				Num_1	Num_2
R2				Num_3	Num_4
R3					
R4					
R5					

[Moving rule]

- The coordinates of the row : The remainder of Num_i divided by 9
- The coordinates of the column : The remainder of Num_i divided by 6
- Find the Num_i whose row and column coordinates match.

- * Find the four integers that fit into the cells.
- * Describe the rules found during the problem solving.
- * Describe how you came to the conclusion.

그림 7. 1차 테스트 문항 : 문제 해결
Fig. 7. 1st test problem : problem solving

2차 테스트의 첫 번째 문항은 그림 8과 같이 75~138 사이의 숫자 64개의 합계를 구하고, 이를 10으로 나눈 나머지를 산출하는 문항이다. 각 셀의 값을 10으로 나눈 나머지를 산출하고 다시 이들의 합계를 10으로 나눈 나머지를 산출한다면 빠르게 답을 산출할 수 있다. 반면, 64개 셀의 모든 값의 합계를 산출한 뒤 10으로 나누는 방식으로 답을 산출한다면 제한된 시간 내에 답을 산출하기 어렵다. 본 문항은 적용할 규칙과 결과를 산출하는 방식을 찾아내는 것이 핵심이다. 그 규칙을 찾아내는 과정에서 패턴 인식 역량과 문제 분해 및 추상화 역량의 증진과정을 파악할 수 있게 설계했다.

[Pro1] In the array below, calculate the remainder by dividing the sum of all numbers by 10.

75	76	77	78	79	80	81	82
83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106
107	108	109	110	111	112	113	114
115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138

- * Calculate the remainder of the sum of all the values in the upper numeric array divided by 10.
- * Describe how you came to the conclusion.

그림 8. 2차 테스트 문항 : 패턴 인식
Fig. 8. 2nd test problem : pattern recognition

2차 테스트의 두 번째 문항은 그림 9와 같이 설계했으며 1차 테스트의 두 번째 문항의 연속이다. 그림 9의 Num_2 가 39라면 7과 6으로 나눈 나머지가 각각 4와 3이므로 4행 3열로 이동할 수 있다. 마찬가지로 53이라면 7과 6으로 나눈 나머지가 각각 4와 5이므로 4행 5열로 이동할 수 있다. 즉, Num_2 는 7의 배수보다 4가 크고, 6의 배수보다 1 또는 3 또는 5 큰 숫자라는 규칙을 만족해야 한다. 이처럼 주어진 규칙을 이해하고 값을 하나씩 찾아야 하는데 이 과정에서 찾아낸 값은 14로 나눈 나머지가 11인 규칙을 갖는다. 즉, 값을 찾지만 해야 하는 문제가 아니라, 찾아낸 값 사이에 또 다른 규칙을 발견해서 절차적으로 기술해야 하는 문항이다. 찾아낸 값 사이의 규칙을 일반화하는 과정에서 추상화 역량을 키우고 이 문항을 알고리즘 설계하는 과정을 살펴봄으로서 알고리즘 역량을 파악할 수 있게 설계했다.

[Pro2] Find four integers that fit into the cell that satisfy the following moving rule:

	C1	C2	C3	C4	C5
R1				Num_1	Num_2
R2				Num_3	Num_4
R3					
R4					
R5					

[Moving rule]

- The coordinates of the row : The remainder of Num_i divided by 7
- The coordinates of the column : The remainder of Num_1 divided by 6
- Find the value where the coordinates of the row and the coordinates of the column are either (4, 1), (4, 3), or (4, 5).

- * Find the four integers that fit into the cells.
- * Describe the rules found during the problem solving.
- * Describe how you came to the conclusion.

그림 9. 2차 테스트 문항 : 문제 해결

Fig. 9. 2nd test problem : problem solving

V. 적용 및 결과

본 연구에서는 4장에서 설명한 바와 같이 DPAA(P) 모델에 기반하여 정보의 처리, 패턴 인식, 알고리즘을 중심으로 한 표 2의 엑셀 기반 컴퓨팅 교육 모형을 설계하였다. 해당 교육 내용으로 비전공자 대상으로 컴퓨팅 수업을 진행하면서 학습자에게 4장에서 개발한 문항으로 테스트를 진행하고 이를 통해 학습자의 패턴 인식, 알고리즘, 추상화 역량이 개선되었는지 확인하였다.

표 3. 실험 대상 소개

Table. 3. Description of experiment group

College	Number of people
Biotechnology and Natural Resources	57
Business & Economics	43
Art	15
Social Sciences	6
etc	4
Total	125

본 연구는 C 대학교 ‘컴퓨팅적사고와 문제해결’ 과목을 수강하는 1학년 SW 비전공 학생 125명을 대상으로 했으며 학습자 구성은 표 3과 같다. 이 중 기타 영역에 속하는 학습자는 인문계열 2명, 자연과학계열 1명, 체육계열 1명이다. 학습자 모두는 소프트웨어와 전공이 관련성이 없거나 적은 계열로 소프트웨어 교육을 필수로 하지 않는다. 학습자에게 각 테스트 결과는 익명으로 처리되며 통계 값만 연구에 활용될 수 있음을 고지했다.

표 4. 학습 및 테스트 일정

Table. 4. Schedule for learning and experimental testing

Week	Subject	Focus
1	Introduction of Computational Thinking	Computational Thinking
2~3	Presentation of Data in Computer	Abstraction
4~5	Input Data/Output Data and Problem Solving by Procedure	Abstraction
6~8	Conditional Formatting and Control Statements in EXCEL	Pattern Recognition
1st Test		
9~12	Control statements and Random in VBA	Programming
13~15	Projects and Algorithms	0Algorithm
2nd Test		

본 연구의 엑셀 기반 컴퓨팅 교육 내용의 효과를 분석하기 위해 표 4에 나타난 바와 같이 학습자를 대상으로 8주차까지 교육한 뒤 1차 테스트를 시행하고 VBA를 이용한 알고리즘의 구현과 프로젝트 교육이 종료된 15주차 이후에 2차 테스트를 시행하였다. 이는 9~15주차 간 진행된 교육의 효과성을 집중적으로 분석하고자 함이다.

표 5. 문항별 대응표본 통계량

Table. 5. The statistical description of samples for each problem

Case		Mean	samples	S.D.	Standard Error Mean
Pro1	1 st test	8.82	125	2.505	.224
	2 nd test	8.39	125	2.808	.251
Pro2	1 st test	6.29	125	3.941	.352
	2 nd test	9.73	125	1.095	.098
Total	1 st test	15.11	125	5.252	.470
	2 nd test	17.88	125	3.395	.304

표 5는 앞서 설계한 연구 방법에 따라 1차 테스트와 2차 테스트의 시행 결과를 각 문항의 성적과 총합 성적에 관한 기술 통계량으로 나타낸 것이다.

패턴 인식에 관한 역량을 확인할 수 있는 문항1은 1차 테스트 결과가 2차 테스트보다 높다. 1차 테스트에서 정확한 답을 찾아낸 사례는 많았으나 답을 찾아낸 과정을 일반화하여 기술하는데 어려움을 겪는 것으로 파악됐다. 반면 2차 테스트의 경우 1차 테스트보다 답을 찾아내는 절차가 복잡하여 정확하지 않은 답을 기술한 사례가 많았으나 알고리즘은 정확하게 기술한 경우가 많았다. 알고리즘 역량을 확인할 수 있는 문항 2는 1차 테스트의 결과보다 2차 테스트의 결과가 높다. 즉 1차 테스트보다 2차 테스트에서 답과 알고리즘을 정확하게 제시한 학습자가 많다는 것을 의미하며 이는 두 번의 테스트 사이에 알고리즘 기반 문제 해결 과정에 집중한 컴퓨팅 사고 교육이 효과가 있음을 시사한다.

엑셀 기반 컴퓨팅 교육을 통해 컴퓨팅 사고력의 요소 중 패턴인식과 알고리즘 관련 역량이 실제로 향상되었는지 확인하기 위해 1차 테스트와 2차 테스트의 성적의 차이를 종속표본 t 검정(Paired T-Test)으로 검증하였다. 테스트 결과의 분석을 위해 SPSS25을 사용하였다. 분석한 결과에 의하면 표 6에 나타난 바와 같이 학습자에 엑셀 기반 컴퓨팅 교육을 시행하였을 때 패턴인식 역량의 향상을 확인할 수 있는 문항 1에 대해서 1차 테스트와 2차 테스트 결과는 t 검정 값은 1.27, 자유도 124에서 유의확률(p 값)이 0.207로 기각할 수 없다. 즉 엑셀 기반 컴퓨팅 수업 과정 전후로 학습자의 패턴인식 역량이 유의미하게 향상되었다고 볼 수 없다. 이는 테스트가 진행된 9~15주차 기간 동안 알고리즘과 프로그래밍에 집중한 교육이 패턴 인식 역량 향상에 영향을 주지 못했음을 의미한다. 반면 알고리즘 관련 역량을 점검하는 문항 2에 대해 같은 방식으로 분석한 결과는 t 검정 값이 -9.86, 자유도 124에서 유의확률(p 값)이 0.000으로 기각역에 해당한다. 이는 엑셀 기반 컴퓨팅 교육 과정이 알고리즘 관련 문항 해결에 영향을 끼쳤으며 t 검정값이 음수이므로 2차 테스트에서 더 긍정적인 결과가 있음을 의미한다.

표 6. 문항별 대응 t-검정의 분석 결과

Table. 6. The analysis results of the paired t-Test for each problem

	1 st test - 2 nd test					t	df	significance (both sides)
Case	Mean	S.D	Standard Error Mean	95% confidence interval of the difference				
				min.	max.			
Pro1	.43	3.80	.340	-.24	1.11	1.27	124	.207
Pro2	-3.44	3.90	.349	-4.13	-2.75	-9.86	124	.000
Total	-2.77	6.10	.545	-3.85	-1.69	-5.07	124	.000

각 문항의 테스트 결과는 서로 다른 의미로 해석될 수 있

나 1차 테스트와 2차 테스트의 총합을 분석한 결과는 유의확률 p 값이 0.000으로 두 결과가 차이가 있음을 확인할 수 있으며 t 값이 -5.07로 음수이므로 2차 테스트에서 학습자가 유의미하게 더 나은 성적을 획득했음을 알 수 있다.

결과적으로 본 연구에서 개발한 엑셀 기반 컴퓨팅 교육 모형과 테스트를 SW 비전공자를 대상으로 적용하였을 때 그 차이가 유의미하게 긍정적인 방향으로 나타났으며, 이는 비전공 대학생들에게 엑셀과 VBA를 활용한 컴퓨팅 교육을 시행함으로써 컴퓨팅 사고력을 향상할 수 있다는 것을 의미한다고 볼 수 있다.

VI. 결 론

본 연구에서는 이수진 외[9]가 설계한 엑셀 기반 컴퓨팅 사고 교육 모형을 정보, 패턴 인식, 알고리즘에 중점을 둔 교육 모형으로 개편하여 C대학 SW 비전공자 대상으로 교육하고, 2회의 테스트 문항을 개발하여 컴퓨팅 교육의 효과성을 분석했다. 테스트는 8주차와 15주차까지 교육한 뒤 진행했으며 패턴 인식과 해당 패턴이 가진 규칙을 알고리즘으로 표현하는 문항으로 구성했다. 8주차에 진행된 테스트와 15주차에 진행된 테스트의 문항 별 성취 수준의 변화를 비교해본 결과 전체적인 성적은 유의미하게 개선되었음을 확인했다. 세부 문항 별로 살펴보면 패턴 인식 역량에 있어서는 8주차 테스트와 15주차 테스트에서 큰 차이를 보이지 않았으나 알고리즘 역량은 유의미한 수준에서 긍정적인 결과를 나타냄을 확인할 수 있었다. 이는 엑셀 기반의 문제 해결 과정을 통해 기본적으로 내재하고 있는 패턴인식 역량을 기반으로 규칙을 찾아내고 알고리즘으로 만들어내는 역량이 개선되었다는 의미로 해석할 수 있다. 나아가 컴퓨팅 교육의 컴퓨팅 사고력 배양의 효과를 측정하기 위해 개발된 문항은 특정 프로그래밍 언어에 의존적이지 않고 일반적인 패턴과 규칙을 발견하는 과정이다. 따라서 본 연구에서 활용한 테스트들은 엑셀 기반 컴퓨팅 교육 과정뿐만 다른 프로그래밍 도구를 활용하는 컴퓨팅 사고 교육 모형에도 적용될 수 있을 것이다. 향후 연구로 컴퓨팅 사고의 학습 효과를 측정하기 위한 테스트를 다양하게 개발하여 스크래치, 파이썬, 랩터 등을 이용한 컴퓨팅 교육 모형에도 적용할 것이다. 본 연구가 컴퓨팅 사고 배양의 효과성을 분석하는 도구 및 방법을 제시하는 것으로 기대한다.

참고문헌

- [1] What is ICT in education ? The meaning of ICT and ICTs in education research [Internet]. Available: https://en.wikibooks.org/wiki/ICT_in_Education/Definition_of_Terms.

- [2] A. Yadav, et al., "Computational thinking in teacher education," in *Emerging research, practice, and policy on computational thinking*, Springer, Cham, pp.205-220, 2017.
- [3] Y. B. Kafai, "From computational thinking to computational participation in K-12 education," *Communications of the ACM*, Vol. 59, No. 8, pp. 26-27, 2016.
- [4] F. J. García-Peñalvo and J. A. Mendes, "Exploring the computational thinking effects in pre-university education," *Computers in Human Behavior*, Vol. 80, pp. 407-411, 2018.
- [5] C. N. Nam and C. W. Kim, "An Analysis of Teaching and Learning Activities in Elementary Mathematics Based on Computational Thinking," in *Journal of The Korean Association of information Education: proceedings*, pp. 47-51, 2011.
- [6] J. E. Nah, "Software Education Needs Analysis in Liberal Arts," *Korean Journal of General Education*, Vol. 11, No. 3, pp. 63-89, 2017.
- [7] S. Y. 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.
- [8] E. C. Papanastasiou and R. E. Ferdig, "Computer use and mathematical literacy: An analysis of existing and potential relationships." *Journal of Computers in Mathematics and Science Teaching*, Vol. 25, No. 4, pp. 361-371, 2006.
- [9] S. J. Lee, M. Lee and Y. Park, "A Study on a Educational Model for Computational Thinking Development with Excel Program," *Journal of Digital Contents Society*, Vol. 20, No. 1, pp. 65-74, 2019.
- [10] S. K. Han, "A Study of a Design-based Coding Education Program for Computational Thinking," *The Journal of Education*, Vol. 39, pp. 169-184, 2019.
- [11] S. Lee, and J. Kim, "Analysis of SW basic education contents for non-majors," in *The Proceeding of Korean Association of Computer Education*, pp. 121-124, August, 2018.
- [12] S. B. Shin, "The Improvement Effectiveness of Computational Thinking through Scratch Education," *Journal of The Korea Society of Computer and Information*, Vol. 20, No. 11, pp. 191-197, 2015.
- [13] Y. Kim and M. Lee, "A Comparative Study of Educational Programming Languages for Non-majors Students: from the Viewpoint of Programming Language Design Principles," *Korean Association of Computer Education*, Vol. 22, No. 1, pp. 47-61, 2019.
- [14] Data Portal [Internet], Available: <https://www.data.go.kr/>
- [15] C. Ware, *Information visualization: perception for design*, 3rd ed. Morgan Kaufmann Publishers, 2015.
- [16] J. S. Sung and H. 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.
- [17] S. Y. Choi, "A Study on Teaching-learning for Enhancing Computational Thinking Skill in terms of Problem Solving", *The Journal of Korean Association of Computer Education*, Vol. 19, No. 1, pp. 52-63, 2016.



박윤수(Yoon-Soo Park)

2014년 : 중앙대학교 전기전자공학부 (공학사)

2016년 : 중앙대학교 전기전자공학과 (공학석사)

2016년~현 재: 중앙대학교 대학원 전자전기공학과 (박사과정)

2016년~현 재: 중앙대학교 다빈치교양대학 시간강사

※관심분야 : SW 교육, 암호학, 타원 곡선 암호, 정보보안, Network



이민정(Minjeong Lee)

1994년 : 중앙대학교 컴퓨터공학과 (공학사)

1996년 : KAIST 전산학과 (공학석사)

1996년~2000년: (주) LG전자 LG종합기술원

2000년~2010년: (주) 아이에이

2011년~2015년: (주) 삼성전자 소프트웨어센터

2016년~현 재: 중앙대학교 다빈치교양대학 조교수

※관심분야 : SW교육, 기계학습, 플랫폼, 에듀테크, AR/VR등



이수진(Sujin Lee)

1996년 : 고려대학교 응용동물과학과 (농학사)

2003년 : 이화여자대학교 디자인학과 (디자인학석사)

2007년 : 서강대학교 미디어공학과 (공학석사)

2013년 : 서강대학교 미디어공학과 (공학박사)

2014년~2015년: 전남대학교 강의

2015년~2017년: 교통대, 한밭대, 협성대, 중앙대, 계원예술대학교 강의

2018년~현재: 중앙대, 세종대학교, 한남대 강의

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