

고등학교 텍스트 기반 프로그래밍 수업을 위한 프레임워크 개발

윤 숙 영¹ · 최 현 중^{2*}¹한국교원대학교 컴퓨터교육과 석사과정^{2*}한국교원대학교 컴퓨터교육과 교수

Developing a Framework for Teaching Text-Based Programming in High School

Soon-Young Yoon¹ · Hyun-Jong Choe^{2*}¹Master's Course, Department of Computer Education, Korea National University of Education, Chungcheongbuk-do, Korea^{2*}Professor, Department of Computer Education, Korea National University of Education, Chungcheongbuk-do, Korea

[요 약]

본 연구는 고등학교에서 텍스트 프로그래밍 학습의 기본 요소를 연계성 있게 가르치고 배우기 위한 프레임워크를 개발하였다. 이 과정에서 첫째, 2015 개정 교육과정과 교과서에 포함된 프로그래밍 영역을 분석하였다. 둘째, 델파이 조사를 통해 텍스트 프로그래밍 학습의 기본 요소의 선정과 조직에 대한 전문가들의 합의를 도출하였다. 셋째, 합의된 결과를 근거로 연계성 있는 텍스트 프로그래밍 학습 프레임워크(CLF-TP)를 제안하였다. 프레임워크의 기본 요소는 학습의 연계성과 계열성을 고려하여 조직되었으며, 고등학교에서 프로그래밍 수업 과정을 결정하는데 활용할 수 있다. 이를 통해 학생들이 프로그래밍 학습을 시작하는 단계에서 체계적으로 개념을 학습할 수 있게 함으로써 추후 소프트웨어 관련 선택 과목을 학습하기 위한 기반을 다질 수 있도록 돕고자 한다.

[Abstract]

This paper develops a framework to support teaching and learning fundamental elements of text-based programming in high school. Foremost, the authors analyzed contents of programming area included in the 2015 revised national curriculum and textbooks. Secondly, experts reached a consensus on the selection and organization of fundamental elements for teaching text-based programming by the Delphi method. Finally, the Continuous Learning Framework for Text-based Programming(CLF-TP) was proposed based on the agreed results. Fundamental elements in the framework were organized in consideration of the continuity and sequence of them. The finding can assist high school teachers to determine the course of their programming classes. It follows that students can build the basic concepts of programming systematically at the beginning of learning so that they can lay the groundwork for learning software-related subjects.

색인어 : 프로그래밍 교육, 텍스트 프로그래밍, 학습 요소, 연계성, 교육과정**Keyword** : Programming education, Text-based programming, Learning elements, Continuity, Curriculum<http://dx.doi.org/10.9728/dcs.2022.23.10.1969>

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 16 September 2022; Revised 07 October 2022

Accepted 20 October 2022

***Corresponding Author; Hyun-Jong Choe**

Tel: +82-43-230-3778

E-mail: chj@knue.ac.kr

1. 서론

IT 기반 사회에서 소프트웨어 교육은 모든 사람들에게 요구되는 기초 교육으로 자리 잡고 있다. 세계 각국은 소프트웨어 교육을 대학뿐 아니라 초·중·고등학교의 기본적인 필수적인 교육으로 강화하고 있다. 영국에서는 기존의 'ICT' 과목을 'Computing' 과목으로 변경하고, 2014년부터 모든 학교급에 적용되는 정규 교과로 편성하였다[1]. 미국에서는 Association for Computing Machinery(ACM)과 Computer Science Teachers Association(CSTA)에서 개발한 컴퓨터 과학(Computer Science) 교육과정을 기반으로 초등학교부터 고등학교까지 단계적이고 유기적인 소프트웨어 교육을 실시하고 있다[2].

지금의 소프트웨어 교육은 컴퓨터를 수동적으로 사용하거나 정보를 검색하는 수준을 뛰어넘어서 학생들이 문제를 해결하기 위해 컴퓨터를 의미 있고 적극적으로 활용하는 문제 해결력을 강조하고 있다[3]. 최근에는 단순히 응용 프로그램을 사용할 수 있는 수준이 아닌 상황에 따라 필요한 프로그램을 직접 만들어 낼 수 있는 제작자 수준까지도 가능하도록 요구되고 있다[4].

프로그래밍은 인간과 컴퓨터가 모두 이해할 수 있는 일련의 명령을 논리적으로 조직하여 컴퓨터에 수행시키는 작업이다[5]. 프로그래밍 교육을 통해 학생들은 정보처리능력 등 다양한 인지적 기능을 향상시킬 수 있다[6]. 프로그래밍 교육은 소프트웨어 교육의 중요한 부분으로 간주되어 미국, 영국뿐 아니라 덴마크, 호주, 프랑스, 핀란드 등 국의 주요 국가들 역시 프로그래밍 중심의 소프트웨어 교육을 국가 교육과정으로 의무화하여 추진하고 있다[7]. 현재 우리나라의 중·고등학교에서 프로그래밍 교육을 위한 대표적인 교과목은 '정보'이다. 그 중 고등학교 '정보'는 일반 선택 과목으로, 중학교 '정보' 교과와 과학 계열 전문 교과 I 과목인 '정보과학'과 연계성을 가지고 있다.

고등학교 정보 교과의 프로그래밍 영역에서는 텍스트 기반의 프로그래밍 언어를 사용하도록 하고 있다. 텍스트 기반 프로그래밍 언어는 프로그래밍에 필요한 명령 구문들을 문자로 입력하는 방식으로, 대표적인 텍스트 기반 프로그래밍 언어로는 C, 파이썬(Python), 비주얼 베이직(Visual Basic) 등이 있다[8]. 중학교에서 엔트리(Entry)나 스크래치(Scratch)와 같은 블록 기반의 프로그래밍 언어를 사용하는 것과 다르게 고등학교에서 텍스트 언어를 활용하도록 하는 이유는 학생들이 자신의 진로와 연계하여 심화된 내용을 학습할 수 있도록 하기 위함이다[9].

고등학교에서 정보 교과는 이후 소프트웨어 관련 과목을 학습하기 위한 선수 과목으로서의 역할을 한다. 따라서 정보 교과에서 배우는 프로그래밍 언어에 대한 기초적인 내용을 잘 학습해야 이를 바탕으로 다른 심화 교과목을 원활히 소화할 수 있다. 그런데 현재 학교 현장에서는 블록 기반의 언어만을 사용해본 학생들이 처음으로 텍스트 프로그래밍 언어를 사용하는 과정에서 학습 부담을 느끼는 경향이 있다. 이 문제를 해결

하는 방법 중 한 가지는 텍스트 프로그래밍 언어를 활용하는데 필요한 가장 기본적인 개념을 연계성 있게 학습할 수 있도록 교육 내용을 구성하는 것이다. 이를 위해서는 우선 교육할 요소를 학생의 수준에 적합하게 선정하고, 이 요소를 학습 순서에 맞게 적절히 조직화하여 교사에게 제시할 수 있어야 한다.

본 연구에서는 현재 교육과정과 교과서의 프로그래밍 영역 구성을 분석한 결과를 토대로 전문가 델파이를 실시하였다. 이를 통해 고등학교 프로그래밍 교육을 위한 기본 요소를 선정하고, 요소 간의 연계성과 계열성을 고려하여 조직하였다. 그 결과를 종합하여 최종적으로는 텍스트 프로그래밍 학습을 위한 프레임워크 CLF-TP(Continuous Learning Framework for Text-based Programming)를 제시하였다. 이를 통해 체계적인 고등학교 텍스트 프로그래밍 수업 내용을 구성하는데 있어 제안점을 도출하고자 하였다.

II. 이론적 배경

2-1 고등학교 프로그래밍 교육 내용

2022 개정 교육과정은 개발 단계에 있으므로 현재 적용되고 있는 2015 개정 교육과정을 중심으로 프로그래밍 교육 현황을 살펴보고자 한다. 2015 개정 교육과정에서 중학교 정보는 '과학·기술·가정·정보' 교과(군)에, 고등학교 정보는 '생활·교양' 교과 영역의 '기술·가정' 교과(군)에 편성되었다. 2015 개정 교육과정에 따른 고등학교 정보 교과의 큰 변화는 일반 선택 과목으로 전환되어 이전보다 많은 학교들이 정보 과목을 선택할 수 있게 되었다는 점이다[10]. 고등학교 정보 교과의 영역은 '정보문화', '자료와 정보', '문제 해결과 프로그래밍', '컴퓨팅 시스템' 네 가지로 구성되어 있다. 그 중 '문제 해결과 프로그래밍'은 컴퓨팅 원리에 따라 문제를 추상화하여 해법을 설계하고, 프로그래밍 과정을 통해 소프트웨어로 구현하여 자동화할 수 있는 능력을 기르는 것을 목표로 한다[11].

현재 프로그래밍 교육의 목표는 자동화 능력을 통해 컴퓨팅 사고력(Computational Thinking)을 함양하는 것이다. 컴퓨팅 사고력이란 해결해야 할 문제를 만났을 때 컴퓨터 과학자처럼 사고하는 것으로, 구체적으로는 추상화와 자동화를 통한 문제 해결 능력을 의미한다[12]. 여기서 자동화는 추상화된 개념이나 절차와 방법 등을 컴퓨팅 기기가 수행할 수 있도록 알고리즘화 하는 과정이다[13]. 즉, 추상화 과정에서 만들어진 모델을 컴퓨터가 이해할 수 있는 프로그래밍 언어로 표현하여 많은 양의 반복된 작업이나 시뮬레이션 하는 것을 의미한다[14]. 따라서 고등학교에서 시행되는 프로그래밍 교육의 목표는 컴퓨팅 사고력의 증진이며, 역으로 프로그래밍은 컴퓨팅 사고력 교육과 가장 직접적으로 관련이 있는 학습 내용이자 방법이라고 할 수 있다. 2015 개정 고등학교 정보 교육과정에서 제시되고 있는 프로그래밍 영역의 내용 요소와 그에 대응되는 학습 요소를 정리하면 <표 1>과 같다.

표 1. 고등학교 정보 교육과정에서 프로그래밍 영역의 내용 요소와 학습 요소

Table 1. Elements of programming in the national informatics curriculum of high school

Content elements	Learning elements
· Program development environment	· Text-based programming environment
· Variables and data types	· Variable · Data type
· Operators	· Arithmetic operators · Comparison operators · Logical operators
· Standard input/output and file input/output	· Standard input/output · File input/output · Design input/output
· Nested control structure	· Nested control structure
· Arrays	· One-dimensional array and Two-dimensional array
· Functions	· Functions
· Programming implementation	· Software development

2015 개정 교육과정에 따른 고등학교 정보 교과서는 2018년에 와서 학교 현장에 보급되어 현재까지 사용되고 있다[15]. 앞에서 언급했듯 고등학교 프로그래밍 영역에서는 텍스트 기반 프로그래밍 언어를 활용하고 있는데, 현재 정보 교과서 8종은 C 또는 파이썬을 실습 언어로 채택하고 있다. 교과서 8종 중 2종(삼양미디어, 금성출판사)은 C를 사용하고 있고, 나머지 6종(천재교과서, 비상교육, 교학사, 씨마스, 교문사, 이오복스)은 파이썬을 사용하고 있다.

먼저 C를 사용하는 교과서 2종은 <표 2>와 같이 ‘문제 해결과 프로그래밍’ 대단원 안에 여러 중단원 및 소단원을 두는 구성을 보이고 있다. 반면 파이썬을 사용하는 교과서 3종은 <표 3>과 같이 ‘문제 해결과 프로그래밍’ 대단원 안에 ‘프로그래밍’ 중단원을 두고, 소단원마다 한두 가지의 요소를 포함하도록 제시하고 있다. 전체적으로 현 교과서들의 프로그래밍 영역 구성은 교육과정에서 제시하는 내용 요소와 학습 요소의 순서를 그대로 제시한 경우도 있으나 순서를 재배치한 경우도 빈번하게 확인할 수 있다.

표 2. C 사용 교과서의 프로그래밍 영역 목차

Table 3. Contents of programming in textbooks using C

	Textbook A		Textbook B	
Chapter	Problem solving and programming			
Unit and Lesson	Unit	Lesson	Unit	Lesson
	How to develop a program	Program development environment	Basic of programming	Program development environment
		C Integrated development environment		
		Starting with C		
	How to program input, store, operate, output of value	Variables and operators	Basic of programming ① (Variables and data types)	
		Input/output		
		Input/output with files		
	How to control the flow of a program	Conditional structure and repetitive structure	Basic of programming ② (Control structures and nested control structures)	
		Programming by conditional structure		
	Why use array and function?	Programming by repetitive structure	Basic of programming ③ (Arrays and functions)	
Arrays				
Function				
Value-returning functions				
How to solve various problem situations through programming	Global variables and local variables	Implementation of programming	Meal voucher Vending machine	
	Example			
	Problem 1			
	Problem 2		Pyramid	

표 3. 파이썬 사용 교과서의 프로그래밍 영역 목차

Table 3. Contents of programming in textbooks using Python

	Textbook A	Textbook B	Textbook C	Textbook D	Textbook E	Textbook F
Chapter	Problem solving and programming					
Unit	Programming					
Lesson	Computer programming	Understanding of programming	Program development environment	programming environment	Fundamental programming	Program development environment
	Developing programs by Python	Data type and operators	Variables and data types	Variables and data types	Input/output	Variables and data types
	Variables and data types	Condition selection and	Standard input/output	Operators	Arrays	Standard input/output & File input/output
	Functions	Repetition	implementing of sequential-repetitive-conditional structure	Standard input/output & File input/output	Control structures	Control structures
	Conditional statement	Arrays and bulk Data	Nested structure	Utilizing control structure	Functions	Arrays
	Repetitive statement	Function modularizing	Functions	Utilizing arrays	Application of Programming	Functions
	Events	Software Development	Lists and arrays	Utilizing functions		
	Input/output function		File input/output	Application of Programming		
	Effective programming					
	Collaborative programming					

2-2 학습 내용의 연계성과 계열성

교육 내용 조직에 있어 연계성(continuity)은 교육제도를 통한 학생들의 효율적 지도와 개발을 위해 교육과정을 비롯한 교육제도 내 여러 요소들을 조화롭게 관련시키는 과정을 의미한다[16]. 연계성에 대한 정의는 교육과정 연구자들 사이에서도 아직 합의되지 못한 부분이 있어 명확하지 않다[16, 17]. 그럼에도 불구하고 연계성은 대체로 기존 교육과정 조직 원리를 모두 포함하는 거대 개념으로 해석되어 사용되고 있다[17]. 한편 연계성보다 구체적인 의미로 사용되는 계열성(sequence)은 교육과정의 누적적인 학습과 학습자의 발달 단계와 학습의 원리를 포함한 계층적인 진행을 의미한다[18].

현재 고등학교 텍스트 프로그래밍 교육을 위한 연계성 있는 교육 내용 선정과 조직에 관한 연구는 발표된 바 없다. 지금까지의 연구는 학교급 간 연계성을 중심으로 발표되고 있어 우선 이에 대해 살펴보고자 한다. 먼저 소미현 외(2016)는 블록 기반 프로그래밍 학습에서 텍스트 기반 프로그래밍 학습으로의 전이 가능성에 대해 연구하여 발표하였다[8]. 김동만 외(2018)는 2015 개정 문제 해결과 프로그래밍 영역의 내용 연계성을 분석하여, 성취 기준의 보완이 필요함을 언급하였다[19]. 전용주(2019)는 초·중등 소프트웨어 교육 요소의 학교급 별 계열성을 분석하고 위계를 탐색하였다[20]. 김한성(2020)은 ACM & IEEE-CS에서 발표한 Computing Curricula 2020: Paradigms for Future Computing

Curricula를 근거로 2015 개정 교육과정 초·중등 학습 요소의 계통적 분석을 시도하였다[21].

한편 타 교과에서는 한 학교급 또는 한 학년의 교육 내용에 대해 위계를 분석하고 이를 바탕으로 한 교육 내용 구성에 관한 연구가 여러 차례 발표되었다. 수학과에서는 우리나라 수학과 교육과정 내용 계통 연구(박윤범 외, 2014)가 발표되었으며, 그 후속 연구로 수학 학습 위계에 따른 수학 평가 보정 모델 개발(허난 외, 2020)이 진행되었다[22][23]. 과학과에서는 지식공간론을 활용한 지구과학 과목에 포함된 천문학적 공간 개념에 대한 위계 및 지식 상태를 분석(윤마병, 2010)이 이루어졌다[24]. 음악과에서는 중학교 장단 학습 위계의 분석적 연구와 이를 바탕으로 한 수업 적용을 시도(윤아영, 2021)하였다[25]. 그 외에도 중학교 3학년 도시 단원에서 지리 학습 개념 위계에 관여하는 요인 탐색에 관한 연구(이형상, 2021) 등 여러 연구가 발표된 바 있다[26]. 따라서 정보교과의 프로그래밍 영역에 대해서도 연계성과 계열성을 고려한 학습 내용 조직에 관한 연구가 필요하다.

III. 연구 설계

3-1 연구 방법

본 연구에서는 고등학교 수준에서 텍스트 프로그래밍의 기초적인 내용을 연계성 있게 학습할 수 있도록 기본 요소를 선

정하고 조직하고자 하였다. 이를 위해 델파이 조사를 통한 전문가 검토 과정을 수립하였다. 델파이 기법은 교육학에서 전문가와 교육구성원의 의견을 수집하고 종합하여 집단적 판단으로 정리하기 위해 많이 사용하는 방법으로 교육 목적 및 목표 설정, 교육과정 개발, 교육 문제 해결, 교수 방법 개발 등 다양한 연구 목적으로 이용되고 있다[27]. 특히 델파이 기법을 활용하면 피상적인 자료 수집에 그칠 수 있는 조사 연구를 피드백의 순환 과정을 거쳐 보완해주는 면이 있다[28]. 이에 본 연구에서는 간소화된 형식의 델파이 조사를 수행하여 전문가들의 의견을 수렴함으로써 보다 타당한 연구 결과를 도출해내고자 하였다. 델파이 조사를 위한 절차는 <표 4>와 같다.

표 4. 델파이 수행 절차

Table 4. The delphi process

Step 1	Analyzing national curriculum and text book
	↓
Step 2	Deciding sample size and selecting expert panel
	↓
Step 3	Developing Delphi round one questionnaire
	↓
Step 4	Releasing and analyzing round one questionnaire
	↓
Step 5	Developing Delphi round two questionnaire
	↓
Step 6	Releasing and analyzing round two questionnaire
	↓
Step 6	Documenting results and suggesting framework

먼저 연구자는 이론적 배경에서 서술한 2015 개정 정보 교육과정의 내용 요소 및 학습 요소와 고등학교 정보 교과서 8종의 프로그래밍 단원 목차를 분석하였다. 또한 정보과학 교육과정 및 교과서(삼양미디어), 프로그래밍 교과서(멘토스쿨)의 구성을 참고 자료로써 함께 검토하였다. 이를 바탕으로 텍스트 프로그래밍 학습 영역을 <표 5>과 같이 구분하고, 고등학생이 프로그램을 작성하며 기본적으로 학습해야 하는 요소를 ‘기본 요소(fundamental elements)’라고 명명하였다. 그리고 이들 기본 요소를 수정해가기 위한 델파이 조사의 방향과 문항의 목적을 수립하였다.

표 5. 프로그래밍 학습 영역

Table 5. Area of learning programming

Meaning	Developing	Evaluating	Practicing
· Concept & Necessary · Language & Development environment	· Fundamental elements · Advanced elements	· Debugging · Efficiency assessment	· Personal project · Collaborative project

델파이 조사에서 패널의 크기는 통계적 검정력에 의존하기 보다는 전문가들의 합의에 도달하기 위한 역동성이 중요하므로 10-18명 정도가 적합하다고 보고된 바 있다[29]. 이에 따라 본 연구의 전문가 패널은 세 집단으로 구분하여 12명을 선정하였다. 첫 번째 집단은 대학교 컴퓨터교육과(인공지능 융합교육전공 포함) 소속 교수 4인이다. 두 번째 집단은 컴퓨터교육(초등컴퓨터교육 포함) 박사 학위 소지자로서, 대학교에서 컴퓨터 교육 관련 강의 경력이 있는 박사 4인이다. 마지막 세 번째 집단은 임용 표시 과목이 정보컴퓨터이며 현직 고등학교 교사로 근무 중인 4인이다. 모든 패널은 컴퓨터 공학 및 교육 경력이 3년 이상인 자료 선정하였다.

이후 선발한 패널을 대상으로 1차 델파이 조사를 실시하였다. 질문지는 파일과 인쇄본을 제공하고 전문가가 선택하여 제출하도록 하였다. 이후 1차 델파이에서 응답한 전문가들의 의견을 종합하여 2차 문항을 설계하여 다시 조사하였다. 이러한 과정을 통해 텍스트 프로그래밍 학습의 기본 요소의 선정과 조직에 관한 전문가들의 합의점을 도출하여 연구 결과로 제시하였다.

3-2 질문지 구성

1) 1차 질문지 문항 구성

1차 델파이 문항은 개방형 질문을 중심으로 구성하되 일부 폐쇄형 질문을 포함하였다. 1차 델파이 문항에 대한 구체적인 정보는 <표 6>과 같다. 총 다섯 개의 문항에 대한 응답을 작성한 후 마지막 장에는 간단한 전문가 인적사항을 기입하도록 하였다.

표 6. 1차 조사의 문항 구성

Table 6. Questions in first survey

Question No	Question types	Question contents
Q.1	Closed-ended	Evaluation Adequacy of word
Q.2	Open-ended	Selection of elements
Q.3	Open-ended	
Q.4	Closed-ended	Relationship between elements
Q.5	Open-ended	Organization of elements
Additional question	Expert profile	

문항 1은 연구에서 사용할 용어를 검증하기 위한 문항으로, 프로그래밍 학습의 ‘기본 요소’라는 용어의 적절성을 리커트(Likert) 5점 척도로 표시하게 하였다. 그리고 만약 응답자가 제시할 만한 용어가 있다면 함께 기입하도록 하였다. 한편 문항 4에서는 프로그래밍 학습의 기본 요소 간 연계성과 계열성에 관해 질문하였다. 이를 통해 전문가들이 기본 요소 간에 연계성과 계열성을 모두 가지도록 구성해야 한다고 생각하는지, 연계성은 가지되 계열성을 가진다 하기는 어렵다고 여기는지

파악하고자 하였다. 해당 두 가지 선택지와 연계성과 계열성 모두 부재, 기타를 포함한 4지 선다형으로 문항을 구성하였다.

나머지 문항 2, 3, 5는 기본 요소의 선정과 조직을 위한 질문으로 모두 개방형 문항으로 제시하였다. 그리고 해당 조사의 목적이 형식의 제한 없이 응답자의 자유로운 의견을 많이 수집하고자 함을 전문가들에게 밝히며, 가능한 구체적인 내용으로 풀어서 응답을 기술할 것을 요구하였다. 문항 2에서는 연구자가 현재 고등학교 교육과정 및 교과서를 참고하여 1차적으로 선별한 후보군을 제시하였다. 그리고 응답자는 각 요소에 대해 기본 요소에 포함, 기본 요소에서 제외, 다른 요소와 통합 중 적절한 방법과 그 이유를 작성하도록 하였다. 문항 3에서는 기본 요소 후보군으로 제시되지 않았으나 응답자가 생각하기에 추가되어야 할 것으로 보이는 기본 요소가 있다면 그 이유와 함께 작성하도록 하였다. 마지막 문항 5에서는 지금까지 응답자가 작성한 내용을 바탕으로 프로그래밍 영역의 학습 내용으로서 조직되어야 한다고 생각하는 전체 기본 요소를 순서대로 나열하고, 그렇게 조직한 이유에 대해 작성하도록 하였다.

2) 2차 질문지 문항 구성

2차 조사에서는 1차 델파이 조사에서 나온 전문가 의견을 종합하여 텍스트 프로그래밍 학습의 기본 요소를 선정 및 조직하여 나열하였다. 그리고 선정된 기본 요소의 적절성과 연계성 및 계열성 있는 학습을 위한 기본 요소 조직의 적절성에 관한 문항으로 구성하였다. 구체적인 2차 조사의 문항은 <표 7>과 같다

표 7. 2차 델파이 문항 구성

Table 7. Questions in second survey

Question No	Question types	Question contents		
		Category		Aspect of adequacy
Q.1-1	Closed-ended	Fundamental elements	Selection	Concept
Q.1-2				Category
Q.2			Organization	Sequence
Q.3-1		Subtypes of fundamental elements	Selection	Concept
Q.3-2				Terminology
Q.4	Organization	Sequence		
Additional question	Any other comments			

수정된 프로그래밍의 기본 요소가 기본 요소와 그 하위 기본 요소(sub-fundamental elements)로 조직되었기 때문에 질문도 이에 따라 구성되었다. 질문 1과 질문 2는 기본 요소의 적절성에 관한 질문이다. 질문 1은 선정된 기본 요소의 적절성에 대해, 질문 2는 기본 요소 조직의 적절성에 대해 평가하도록 하였다. 질문 3과 질문 4는 하위 기본 요소의 적절성

에 관한 질문이다. 질문 3은 선정된 하위 기본 요소의 적절성에 대해, 질문 4는 하위 기본 요소 조직의 적절성에 대해 평가하도록 하였다. 추가 질문을 제외한 2차 조사의 모든 문항은 리커트 5점 척도에 따른 폐쇄형 문항으로 제시하였다. 단 모든 문항마다 기타 의견란을 두어 개선 의견이나 기타 의견이 있다면 작성할 수 있도록 하였다.

IV. 연구 결과

4-1 질문지 응답률

델파이 조사 결과 전문가 패널의 질문지 응답률은 <표 8>과 같다. 최종 선정된 패널 12명 중 11명이 최종 응답하여 약 91.67%의 응답률을 확보하였다.

표 8. 델파이 질문지 응답률

Table 8. Response rate of the delphi process

Expert panels	Size of the panel	Number of respondents	
		Round 1	Round 2
Professor	4	3	3
Post-doctoral researcher ^a	4	4	4
Teacher	4	4	4
Total (Response rate)	12 (100%)	11 (91.67%)	11 (91.67%)

^a Some of the post-doctoral researchers were appointed professors after the survey period(2022.6-7).

델파이 조사 결과의 타당도를 분석하기 위한 내용 타당도 비율(CVR; Content Validity Ratio)은 참여한 패널 수에 따라 그 최솟값이 값이 결정된다. 본 연구에서는 2차 조사까지 참여한 패널이 11명이므로 5점 척도 문항에서 Lawshe(1975)[30]가 제안한 CVR 최솟값 비율 .59를 적용하여 그 기준 이하의 경우 내용타당도가 낮은 문항으로 판정하고자 하였으나 실제 CVR 값이 최솟값에 미치지 못하는 문항은 존재하지 않았다. 따라서 1차 델파이 문항에서 개방형으로 작성한 전문가들의 의견을 종합하여 2차 델파이 문항에서 제시하고, 2차 응답에서 개선된 추가적인 의견을 반영하여 최종 수정 후 확정하는 방법으로 진행되었다.

4-2 1차 조사 결과

1차 델파이 조사 결과 프로그래밍 학습의 ‘기본 요소’라는 용어의 적절성을 묻는 문항 1에 대해 전문가 11인 중 9인이 ‘매우 적절하다’ 또는 ‘적절하다’고 응답하였다. 기타 의견으로 ‘핵심 요소’, ‘기본 문법’ 등도 제시되었으나, 응답 결과 CVR 값 .636, 평균값 3.91, 표준편차 1.13으로 연구자가 제시한 용어에 대해서도 전문가들의 동의가 이루어졌다고 판단

하여 사용하는 용어를 유지하기로 결정하였다. 또한 프로그래밍 학습의 기본 요소 간 연계성과 계열성을 묻는 문항 4에 관해서는 11인 중 9인의 전문가가 요소 간 계열성과 연계성이 모두 있다고 응답하였다. 다른 1인의 전문가는 연계성만 있다고 하였으며, 나머지 1인은 기타로 응답하고 의견을 보류하였다. 따라서 텍스트 프로그래밍 학습의 기본 요소는 연계성을 가지고, 그보다 좁은 의미인 계열성도 일정 수준 이상 가진다는 결론을 도출하였다.

문항 2에서는 기본 요소의 후보군을 제시하고 해당 요소에 대한 전문가들의 의견을 개진하도록 하였다. 후보로 제시된 기본 요소와 이에 대한 전문가들의 의견을 정리하여 나타내면 <표 9>와 같다. 응답 결과에 따르면 ‘파일 입출력’은 삭제해야 한다는 전문가의 의견이 많았으며, 기본 요소로 제시된 요소 중 유일하게 삭제 또는 통합해야 한다는 응답이 포함해야 한다는 응답보다 많은 것으로 나타났다. 그 근거로는 파일 입출력이 입출력의 특수한 형태로 볼 수 있고, 고등학생 수준에서 기본 요소라기보다는 심화 요소로 보는 것이 적합하다는 응답이 있었다. 한편, ‘자료형’과 ‘리스트’의 경우 다른 요소에 비해 통합하는 것이 적절하다는 의견이 많았는데, 이때 자료형은 변수와 통합하고 리스트는 배열과 통합하는 것을 제안하였다.

표 9. 프로그래밍 학습의 기본 요소 후보군에 대한 응답 결과
Table 9. Result of responses to fundamental element options of learning programming

Proposed list of fundamental elements	Expert opinions			Total
	Include	Delete	Integrate	
Variable	9	0	2	11
Data type	6	0	5	11
Operator	10	0	1	11
Standard input/output	10	0	1	11
File input/output	5	4	2	11
Control structure	10	0	1	11
Nested control structure	6	0	5	11
Array	8	0	3	11
List	5	1	5	11
Function	11	0	0	11

문항 3의 응답으로서 추가해야 할 기본 요소로 ‘문제 해결’, ‘프로그래밍 개념’, ‘프로그래밍 흐름’, ‘객체’, ‘라이브러리’, ‘라이브러리와 모듈’, ‘클래스와 객체’, ‘포인터’가 제시되었다. 현재 교과서에서 사용하는 언어를 고려해봤을 때 그 중 포인터는 C언어에서, 객체는 파이썬에서 중요한 개념이기에 언급된 것으로 파악되었다. 그러나 프로그래밍 학습의 일반적인 기본 요소를 선정하고 조직하고자 하는 본 연구의 목적을 고려하여 특정 언어에 치우친 요소는 제외하는 것으로 결정

하였다. 다만 모든 언어에 공통으로 적용되며 여러 전문가에게 반복하여 제시된 코드 작성 시작 방법에 관한 기본 요소와 모듈화 관점의 기본 요소는 각각 첫 번째와 마지막 위치에 추가하기로 하였다.

마지막 문항 5에서 기본 요소의 조직에 관하여 각 전문가가 제시한 순서는 매우 다양한 형태로 제시되었다. 이를 한 가지로 수렴하기 매우 어려웠기에 연구자 2인은 여러 차례 협의를 통해 전문가들이 제시한 순서를 기본 요소를 기본 요소와 그 하위 요소의 2단계 형태로 정리하기로 하였다. 그 결과 <표 10>과 같이 기본 요소가 수정되었다. 기본 요소는 프로그램 작성 기초, 데이터 저장, 데이터 연산, 제어 구조, 프로그램 모듈화의 5가지이며 각 하위 요소를 가지는 형태로 조직하였다.

표 10. 프로그래밍 학습의 기본 요소 수정안
Table 10. Revised fundamental elements of learning programming

Fundamental elements	Subtypes of fundamental elements
Basic of programming	Code and comment
	Standard output
Storing data	Standard input
	Variables and data types
	Assignment operation
	Array and list
Operating data	Arithmetic operation
	Increment/decrement operation
	Relational operation
	Logical operation
	Bitwise operation
Control structure	Sequential structure
	Conditional structure
	Repetitive structure
	Nested control structure
Modular programming	Module and function
	Library

4-3 2차 조사 결과

2차 조사 문항의 통계 분석 결과는 <표 11>과 같다. 모든 문항의 CVR 값이 .59 이상의 값을 보이므로 타당한 안으로 판단할 수 있다. 또한 변동계수(CV)가 .5 이하로 나타나 안정도가 확보되었으므로 델파이 조사를 종료하였다[31]. 그러나 기본 요소에 관한 문항보다 하위 기본 요소의 선정과 조직에 관한 문항이 상대적으로 평균값이 낮은 것을 확인하였다. 따라서 해당 문항에 대해 서술형 응답으로 제시한 전문가 의견을 반영함으로써 그 내용을 보완하고자 하였다.

표 11. 프로그래밍 학습의 기본 요소 수정안에 대한 응답 결과
Table 11. Result of responses to the revised fundamental elements of learning programming

Question No	M	SD	CV	CVR
Q.1-1	4.55	.522	.114	1
Q.1-2	4.55	.688	.151	.818
Q.2	4.73	.467	.098	1
Q.3-1	4.45	.688	.154	.818
Q.3-2	4.45	.820	.184	.636
Q.4	4.09	1.221	.298	.636

하위 기본 요소에 대해 대표적인 전문가 의견으로는 첫째, 데이터 저장의 하위 요소에 관한 내용이 있었다. 한 전문가는 ‘표준 입력’이 변수를 학습한 이후에 학습하도록 하는 것이 학습 순서상 자연스럽다고 주장하였다. 한편, ‘대입 연산’을 자료의 저장 방법으로써 대입을 이해할 수 있도록 용어를 개선해야 한다는 의견도 있었다. 따라서 최종안에서는 두 가지 하위 요소를 하나로 통합하여 ‘입력과 대입’으로 수정하여 ‘변수와 연산자’ 다음 위치에 두었다.

둘째, 데이터 연산의 하위 요소에 관한 전문가 의견이 있었다. 파이썬의 경우 별도의 증감 연산자가 존재하지 않는다는 점을 지적하며, 동시에 이것이 기본 요소로 제시되는 것보다는 필요한 언어에 따라 적절하게 더하여 학습하도록 하는 것이 적절하다는 의견이었다.

표 12. 프로그래밍 학습의 기본 요소 최종안
Table 12. Confirmed fundamental elements of learning programming

Fundamental elements	Subtypes of fundamental elements
Basic of programming	Code and comment
	Standard output
Storing data	Variables and data types
	Input and assignment
	Array and list
Operating data	Arithmetic operation
	Relational operation
	Logical operation
	Bitwise operation
Control structure	Sequential structure
	Conditional structure
	Repetitive structure
	Nested control structure
Modular programming	Module and function
	Library

연구자는 해당 의견이 합리적인 주장이라고 판단하여 증감 연산은 연산자의 하위 기본 요소에서 제외하였다. 이러한 수정 사항을 반영한 프로그래밍 학습의 기본 요소의 최종안은 <표 12>와 같다.

4-4 텍스트 프로그래밍 요소 학습 프레임워크 제안

[그림 1]은 전문가 검토 결과를 바탕으로 설계한 프레임워크 CLF-TP를 나타낸 것이다. CLF-TP에는 5가지 기본 요소와 15가지의 하위 기본 요소가 표현되어 있다. 기본 요소는 프로그램 작성 기초, 데이터 저장, 데이터 연산, 제어 구조, 프로그램 모듈화이며 각각 2, 3, 4, 4, 2가지 하위 기본 요소를 포함하고 있다. 하위 기본 요소를 나열하면 코드와 주석, 표준 출력, 변수와 자료형, 입력과 대입, 배열과 리스트, 산술 연산, 관계 연산, 논리 연산, 비트 연산, 순차 구조, 선택 구조, 반복 구조, 중첩 제어 구조, 모듈과 함수, 라이브러리이다.

텍스트 프로그래밍 학습의 기초 단계에서는 선정된 기본 요소를 단계적으로 학습해 나간다. 기본 요소를 연계성 있게 학습해 가는 과정에서는 새롭게 학습하는 기본 요소뿐만 아니라 이전 단계에서 학습한 내용에 대한 이해도를 함께 높일 수 있다. 기본 요소는 텍스트 프로그래밍을 학습하는 시작 단계에 반드시 학습해야 하는 내용들로 구성되어 있다. 따라서 기본 요소 학습을 마친 뒤에는 자료구조 및 알고리즘 구현이 포함되는 심화 요소로 학습을 이어 진행할 수 있다. 이러한 텍스트 프로그래밍 학습의 요소가 구체적인 활동 및 내용으로 제시될 때는 학습에 필요한 기능적 측면과 가치적 측면이 함께 작용하게 된다.

프레임워크는 고등학교에서 텍스트 프로그래밍 학습 과정을 결정하는데 활용될 수 있다. 예를 들어 교사가 배열과 리스트에 대한 수업을 설계하고자 할 때 다음과 같은 사항을 고려할 수 있다. 먼저 배열과 리스트를 데이터 저장의 하위 요소로서 학습하는 것이므로 단순 사용법이 아닌 데이터 저장 방법의 연속선상에서 이해할 수 있도록 수업을 구상한다. 활동 내용이나 동기 유발은 이전에 입력과 대입에서 학습한 내용을 바탕으로 결정하며, 추후 자료의 연산에서 학습할 내용과 어떻게 연계할 수 있을지 고려할 수도 있다. 차시 수업을 구체적인 학습 목표로 진술할 때는 기본 요소를 통해 지식적 측면을 표현하고 이에 따른 기술적, 가치적 측면을 결정하여 표현할 수 있다.

V. 결론 및 제언

본 연구에서는 현재 교육과정 및 교과서를 대상으로 문헌 조사를 실시하고, 조사한 내용을 토대로 델파이 조사를 실시하였다. 이를 통해 고등학교 프로그래밍 수업에서 포함해야 할 기본 요소의 선정과 조직에 대한 전문가들의 합의를 도출하였다.

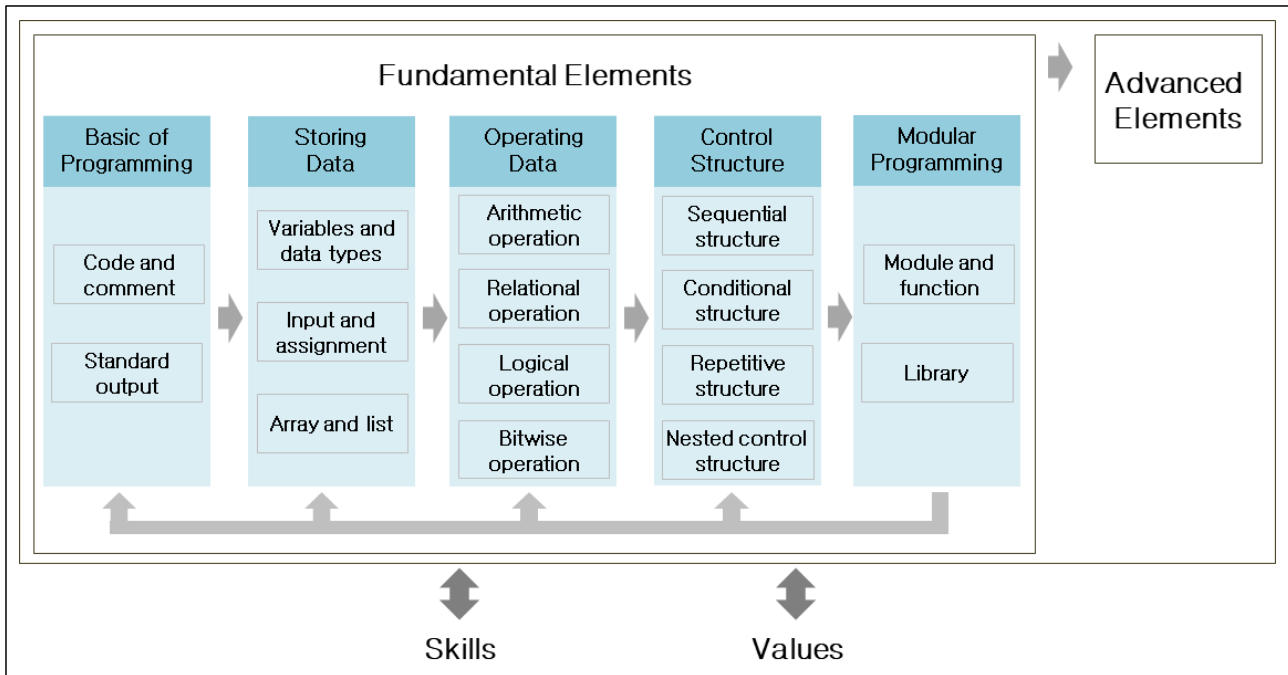


그림 1. 연계성 있는 텍스트 프로그래밍 학습 프레임워크(CLF-TP)

Fig. 1. Continuous Learning Framework for Text-based Programming(CLF-TP)

그 결과를 종합하여 최종적으로 텍스트 프로그래밍 수업을 위한 프레임워크로서 CLF-TP를 제시하였다. 해당 프레임워크가 가지는 의의는 텍스트 프로그래밍을 처음 경험하는 고등학생들이 반드시 배워야 하는 기본 요소를 선정하고 그 연계성을 고려하여 조직했다는 점이다. 이를 통해 학생들이 프로그래밍 학습 과정에서 느끼는 어려움은 덜고 추후 심화 학습을 계속해나가는 발판을 제공하고자 하였다.

CLF-TP는 텍스트 프로그래밍 수업 내용과 활동을 구상하는데 활용할 수 있다. 나아가 한편으로는 기본 요소 간에 연계성뿐만 아니라 계열성을 보인다는 전문가 의견을 교육과정 구성 측면에서 적극적으로 검토해볼 필요가 있다. 즉 기본 요소의 학습 순서를 한 수업에서 선형적으로 배치되는 것보다 중학교 또는 고등학교의 여러 학년에 걸쳐 나선형 교육과정으로 설계되는 것이 바람직할 수도 있다. 따라서 이러한 텍스트 프로그래밍 교육을 위한 효과적인 교육과정 설계에 대한 연구를 본 연구의 후속 연구로서 제안한다.

또 다른 후속 연구로 본 연구에서 제시한 기본 요소를 연계성 있게 학습할 수 있는 문제 해결 학습 모델에 대한 개발을 제안한다. 현재 프로그래밍 수업에서는 문제 기반 학습(PBL; Problem Based Learning), 창의적 문제 해결법(CPS; Creative Problem Solving) 등을 적용했을 때 높은 효과를 보이는 것으로 나타났다[32]. 소프트웨어 교육을 위한 모델로 NDIS(Needs-Design-Implementation-Share), DPAA P(Decomposition-Pattern Recognition-Abstraction-Algorithm-Programming) 등도 고안되었다[33]. 이러한 모델들의 특징은 기초 요소에 대한 학습이 일정 수준 이상 완료된

후, 프로젝트나 문제 해결을 수행하는 것을 전제하고 있다는 점이다. 그러나 실제 고등학교 수업 현장에서는 많은 시간을 기본 개념 이해와 습득에 할애하고 있다. 만약 프로그래밍 수업이 문제 해결을 위한 사고 과정 없는 프로그래밍 언어 교육으로 초점이 맞춰진다면 학생들은 프로그래밍 교육이 어렵고 자신과 관련 없는 교육으로 인식할 위험성이 있다[34]. 따라서 프로그래밍에 필요한 기본적인 학습을 진행하는 과정에서도 문제 해결력을 함양할 수 있도록 하는 교수학습 모델에 대한 연구가 이뤄져야 할 것이다.

소프트웨어 교육은 이제 선택이 아닌 필수이다. 본 연구를 비롯하여 더 좋은 고등학교 프로그래밍 수업을 만들기 위한 연구들이 계속되기 바란다. 이를 통해 정보 수업 시간이 학생들의 컴퓨팅 사고력을 키워주는 결정적 순간이자 자신의 진로를 향해 나아가는 기회로 거듭날 수 있기를 소망한다.

참고문헌

[1] H. R. Kim, The “Research about Policy Background of Computer Science Education in UK School: Lesson from the UK”, *Journal of the Korean Association of Information Education*, Vol. 20, No. 2, pp. 207-218, April 2016. <https://doi.org/10.14352/jkaie.20.2.207>

[2] Y. S. Joeng, S. B. Shin and Y. H. Sung, “Analysis of Appropriateness in Information Curriculum for Algorithm and Programming Education”, *Journal of the Korean*

- Association of Information Education*, Vol. 20, No. 6, pp. 575-584, December 2016.
<https://doi.org/10.14352/jkaie.2016.20.6.575>
- [3] S. K. Jeon, K. A. Sang and S. W. Park, "Analysis of the Current Status of Computer and Information Literacy Education in middle school students", *The Journal of Korean Association of Computer Education*, Vol. 22, No. 2, pp. 11-20, February 2019.
<https://doi.org/10.32431/kace.2019.22.2.002>
- [4] J. S. Sung and H. C. 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 January 2015.
<https://doi.org/10.32431/kace.2015.18.1.005>
- [5] H. S. Choi, "Domestic Literature Review on Computational Thinking Development through Software Programming Education", *Journal of Educational Technology*, Vol. 34, No. 3, pp. 743-774, September 2018,
<https://doi.org/10.17232/KSET.34.3.743>
- [6] Y. S. Han, "Effectiveness of problem-based learning based programming education : Focus on Computational Thinking", *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, Vol. 8, No. 7, pp. 433-445, July 2018.
<https://dx.doi.org/10.21742/AJMAHS>
- [7] E. K. Lee, "Perspectives and Challenges of Informatics Education: Suggestions for the Informatics Curriculum Revision", *The Journal of Korean Association of Computer Education*, Vol. 21, No. 2, pp. 1-10, March 2018.
- [8] M. H. So and J. M. Kim, "Transference from learning block type programming to learning text type programming", *The Journal of Korean Association of Computer Education*, Vol. 19, No. 6, pp. 55-68, November 2016.
<https://doi.org/10.32431/kace.2016.19.6.006>
- [9] W. Y. Chang and S. S. Kim, "A review on trends of programming(algorithm) automated assessment system and it's application", *The Journal of Korean Association of Computer Education*, Vol. 20, No. 1, pp. 13-26, January 2017. <https://doi.org/10.32431/kace.2017.20.1.002>
- [10] H. J. Choe and T. W. Lee, "A Study on the Development and Implementation of Computational Thinking Education Framework", *Journal of The Korea Society of Computer and Information*, Vol. 21 No. 9, pp. 177-182, September 2016, <https://dx.doi.org/10.9708/jksci.2016.21.9.177>
- [11] Ministry of Education Notice No. 2015-74 [Annex 10], Ministry of Education, September 2015
- [12] J. M. Wing, "Computational Thinking", *Communications of the ACM*, Vol. 49, No. 3, pp. 33-35, March 2006.
<https://doi.org/10.1145/1118178.1118215>
- [13] S. Y. Choi, "An Analysis of 'Informatics' Curriculum from the Perspective of 21st Century Skills and Computational Thinking", *The Journal of Korean Association of Computer Education*, Vol. 14, No. 6. pp. 19-30, November 2011.
<https://doi.org/10.32431/kace.2011.14.6.003>
- [14] T. W. Lee, H. J. Choe and Y. J. Jeon, *Informatics education*, 2th ed. Seoul, Hanbit Academy, pp. 197, 2020.
- [15] S. H. Kim, "Analysis of Abstraction Contents in Informatics Textbooks of Middle School According to 2015 Revised Curriculum", *The Journal of Korean Association of Computer Education*, Vol. 21, No. 4. pp. 1-10, November 2011.
<https://doi.org/10.32431/kace.2018.21.5.001>
- [16] S. H. Song, Y. H. Lee and M. O. Kim, "A Curricular Articulation Analysis on the Probability and Statistical Subjects in the Mathematics Textbooks from Elementary School to High School", *Journal of Korea Society of Mathematics Education*, Vol. 28, No. 1, pp. 13-27, June 1989.
- [17] Policy Review No. 2016-15, Gyeonggi Institute of Education, December 2016.
- [18] J. M. Kim, H. S. Woo, H. J. Yang, M. J. Kim, S. H. Kim, S. Y. Yi, B. S. Kim, Y. A. Kim, J. H. Gwak, H. J. Choe, I. K. Jeong, Y. J. Lee and W. G. Lee, "Proposing the informatics standard curriculum scheduled to be revised in 2022", *The Journal of Korean Association of Computer Education*, Vol. 23, No. 1, pp. 1-28, January 2020.
<https://doi.org/10.32431/kace.2020.23.1.001>
- [19] D. M. Kim and T. W. Lee, "An Analysis of the Content Interrelationship in the Problem Resolution and Programming Area On 2015 revised Informatics curriculum", in *Proceeding of the 2018 Winter Symposium in Korean Association of Computer Education*, Vol. 23, No. 1, pp. 115-116, December 2018.
- [20] Y. J. Jeon. "Sequence Analysis and Suggestion for Linking Contents of Elementary and Secondary Software Education", *Journal of Creative Information Culture*, Vol. 5, No. 2, pp. 105-116, August 2019.
<https://doi.org/10.32823/jcic.5.2.201908.105>
- [21] Direction of K-12 SW·AI Education through Computing Curricula 2020, Software Policy & Research Institute, Issue Report IS-110, December 2020.
- [22] A study of the hierarchical structure of the content in school mathematics curriculum", Korea Foundation for the Advancement of Science and Creativity, Research Report 2014, December 2014.
- [23] N. Huh, H. Kim and S. C. Kim, "An Application of Mathematics Assessment and Correction Model According

to Learning Hierarchy”, *Journal of Korea Society Educational Studies in Mathematics <School Mathematics>*, Vol. 22, No. 3, pp. 671- 687. September 2020. <https://doi.org/10.29275/sm.2020.09.22.3.671>

[24] M. B. Yoon, “Hierarchical Analysis of Astronomical Concepts Using the Knowledge Space Theory”, *Journal of Korean Earth Science Society*, Vol. 31, No. 3, pp. 259–266, June 2010, <https://doi.org/10.5467/JKESS.2010.31.3.259>

[25] A. Y. Yoon, “An analytical study on the hierarchy of jangdan learning for integrated education and application of classes: Focusing on secondary music textbooks of 2015 revised curriculum”, *Journal of Learner-Centered Curriculum and Instruction*, Vol. 21, No. 21, pp. 841-855, October 2021. <https://doi.org/10.22251/jlcci.2021.21.21.841>

[26] H. S. Lee, “A Study on the Exploration of Factors Involving Conceptual Hierarchy in Geography Learning: Focused on Urban Chapter in Middle School Social Studies”, *Journal of Learner-Centered Curriculum and Instruction*, Vol. 21, No. 13, 2021, pp. 653-668, July 2021. <https://doi.org/10.22251/jlcci.2021.21.13.653>

[27] J. S. Lee, *Delphi method*, Seoul, Kyoyookbook Press, 2001.

[28] J. R. Kim and J. E. Bae, “A Delphi study on elementary school English education for the 1st & 2nd graders. English Teaching”, *Korea Association Of Teachers Of English*, Vol. 62, No. 4, pp. 487-511, November 2007. <https://doi.org/10.15858/engtea.62.4.200712.487>

[29] C. Okoli and S. D. Pawlowski, “The Delphi method as a research tool: an example, design considerations and applications”, *Journal of Information & Management*, Vol. 42, No. 1, March 2004. <https://doi.org/10.1016/j.im.2003.11.002>

[30] C. H. Lawshe, “A quantitative approach to content validity”, *Personnel Psychology*, Vol. 28, No. 4, pp. 563–575. December 1975. <https://doi.org/10.1111/j.1744-6570.1975.tb01393.x>

[31] J. M. English and G. L. Kernan, “The prediction of air travel and aircraft technology to the year 2000 the Delphi method”, *Transportation research*, Vol. 10, No.1, pp. 1-8, February 1976. [https://doi.org/10.1016/0041-1647\(76\)90094-0](https://doi.org/10.1016/0041-1647(76)90094-0)

[32] S. K. Jeon and Y. J. Lee, “Meta-analysis of the programming learning effectiveness depending on the teaching and learning method”, *The Journal of Korean Association of Computer Education*, Vol. 22, No. 11, pp.125-133, November 2017. <https://doi.org/10.9708/jksci.2017.22.11.125>

[33] Research on SW Education Model Development, Korean educational development institute, CR 2015-35, December 2015

[34] H. S. Kim and B. H Yun, “A Study on Teaching-Learning Model Development for Problem Solving based on Computational Thinking,” *Journal of Digital Contents Society*, Vol. 21, No. 5, pp. 865-873, May 2020. <https://doi.org/10.32431/kace.2019.22.6.005>

윤숙영(Sook-Young Yoon)



2017년 : 성균관대학교 컴퓨터교육과 (이학사), 경영학과 (경영학사)

2018년~현재 : 인천진산과학고등학교 교사
2021년~현재 : 한국교원대학교 컴퓨터교육전공 석사과정
※ 관심분야 : 컴퓨터교육, 융합교육, 정보영재교육 등

최현종(Hyun-Jong Choe)



2005년 : 한국교원대학교 컴퓨터교육전공 (교육학 박사)

2006년~2021년: 서원대학교 컴퓨터교육과 교수
2021년~현재 : 한국교원대학교 컴퓨터교육과 교수
2021년~현재 : 한국교원대학교 정보교육연구소 소장
2021년~현재 : 한국교원대학교 SW·AI연구소 소장
※ 관심분야 : 컴퓨터교육학, 정보교과교육, 인공지능교육 등