

중학교 정보 교과서의 학습 문제 분석 : 문제해결과 프로그래밍 단원을 중심으로

Analysis of Learning Problems in Middle School Informatics Textbooks : Focusing on Problem Solving and Programming Chapter

김슬기[†]
Seulki Kim[†]

요 약

본 연구는 디지털 대전환 시대의 필수 역량인 컴퓨팅 사고력 개발에 초점을 맞춰 문제 해결 경험의 중요성을 확인하고 2015 개정 교육과정 중학교 정보 교과서의 문제해결과 프로그래밍 단원 속 학습 문제를 문제의 유형, 프로그래밍 교육 순서, 맥락의 관점에서 분석하였다. 분석 결과 교과서에서 제시되는 학습 문제들은 비교적 구조성이 높은 형태가 많았으며 프로그래밍 교육 순서 측면에서 각 순서가 고른 비율로 순차적으로 제시되고 있음을 확인할 수 있었다. 맥락 측면에서는 분류되지 않는 문제가 가장 많이 나타났고 개인적 맥락의 문제 비중이 비교적 높게 나타났다. 학생들의 컴퓨팅 사고력을 길러주기 위한 학습 문제 분석의 시사점으로는 구조성이 낮은 문제를 포함하는 다양한 유형의 학습 문제 제시와 프로그래밍 개념 학습 시 문제 해결 경험을 통해 개념을 내재화 할 수 있는 교수학습 환경 구성이 필요함을 확인하였다. 또한 학습 문제의 맥락의 측면에서 개인적 맥락을 포함한 다양한 맥락의 문제 해결 경험이 제시되어야 함을 도출하였다. 본 연구의 결과가 새롭게 시작될 2022 개정 교육과정의 정보 교육을 위한 학습 문제 선정에 기초 연구로 활용되어 교육 현장에 도움을 줄 수 있기를 기대한다.

주제어: SW·AI교육, 컴퓨팅 사고력, 학습 문제, 정보 교과, 교과서 분석

ABSTRACT

This study focuses on the development of computational thinking, a crucial skill in the era of digital transformation. It examines the significance of problem-solving experiences and analyzes the problem-solving and programming units in the revised 2015 middle school informatics textbooks. The analysis categorizes problems based on their type, programming instruction sequence, and context. The findings reveal that the problems presented in textbooks are predominantly structured, and the sequencing of programming education is evenly distributed across different stages. Regarding context, the most frequent were unclassified problems, with a relatively high proportion of personal-context problems. Implications for enhancing students' computational thinking include the need for a diverse range of problem types, including those with less structure, and an instructional environment that fosters conceptual internalization through problem-solving experiences. Furthermore, it is suggested that problem-solving experiences should cover various contexts, including personal ones. This study aims to contribute to the selection of learning problems for the upcoming 2022 revised curriculum in informatics, providing foundational research to assist educational problems.

Keywords: SW·AI Education, Computational Thinking, Learning Problems, Informatics Subject, Textbook Analysis

1. 서론

최근 발표된 2022 개정 교육과정 총론에서는 미래

의 새로운 기술과 사회의 변화에 적응하며 미래사회를 살아갈 학생들이 주도적으로 삶을 이끌어 갈 수 있는 능력 함양을 구성 중점 중 하나로 언급하고 있으며

[†]정회원: 안산원곡초등학교 교사

학습의 기초인 언어, 수리와 함께 디지털 기초 소양을 강조하고 있다[1].

특히 정보 교과 교육과정은 디지털 대전환 시대의 국가와 사회의 요구 사항을 반영하여 미래 사회 변화에 적극적으로 대응할 수 있는 역량을 강화하기 위한 방향으로 설계되었다. 이에 정보 교과 역량으로 컴퓨팅 사고력과 디지털 문화 소양, 인공지능 소양을 설정하였으며 2015 개정 교육과정에서도 강조되고 있는 컴퓨팅 사고력을 바탕으로 실생활 및 다양한 학문 분야의 문제를 해결하는 능력과 태도를 기르는 교과로 그 성격을 정의하고 있다[1, 2].

정보 교과서를 다각도로 분석한 선행 연구들에서는 학생들의 미래 역량을 길러 주기 위해 자료를 수집하고 분석하며 지식을 구성할 수 있도록 탐구적인 교과서의 구성이 강조되었으며 이를 위해 정보 교과서의 교육 목표를 분류하거나 탐구성을 분석하여 교과서의 구성 및 제시된 개념과 표현에 대한 연구가 주로 이루어졌다. 특히 Romney 분석 방법과 Anderson 교육 목표 분류 방법을 사용하여 정량적으로 분석한 연구가 다수 이루어졌으며 교육 현장에 많은 도움을 주었다 [3-6].

컴퓨팅 사고력의 정의와 함께 정보 교과는 학생들의 창의적 문제 해결력의 증진을 중요한 목표로 삼고 있다. 하지만 학생들이 직접 탐구하는 대상이며 교수 학습 활동 중 접하게 되는 학습 문제에 대한 체계적인 연구는 상대적으로 부족한 실정이다[7-9].

이에 본 연구는 컴퓨팅 사고력을 중심으로 미래를 살아갈 학생들의 역량을 효율적으로 증진하는데 도움을 주고자 학생들이 교수학습 과정에서 문제 해결의 원리를 익히고 프로그래밍을 통해 실제로 문제를 해결해보는 경험을 갖게 되는 2015 개정 중학교 정보 교과서의 문제해결과 프로그래밍 단원의 학습 문제를 다각도로 심도 있게 분석하여 시사점을 도출하고 다가올 2022 개정 정보 교육과정에 적합한 학습 문제 구성을 제안하고자 한다.

2. 연구 배경

2.1 정보 교과서 분석 선행 연구

2015 개정 정보 교육과정과 2022 개정 정보 교육과정은 개정 및 발표 시기의 시간적 차이가 있지만 공통적으로 디지털 대전환 시대의 국가적, 사회적 요구를

반영하여 컴퓨팅을 활용한 문제 해결을 전제로 실생활과 다양한 학문 분야의 문제를 해결할 수 있는 컴퓨팅 사고력을 핵심 역량으로 강조하고 있다[1, 2].

교육과정이 학습자의 역량을 길러 줄 수 있도록 교육 내용과 교육 방향을 체계적으로 정리하여 제시한 것이라면 교과서는 교육과정의 내용에 의거하여 교육이 수행될 수 있도록 학습에 도움을 주는 가장 중요한 요소 중 하나라고 할 수 있다[3].

이에 정보 교과의 역량인 컴퓨팅 사고력을 길러주기 위한 관점에서 교과서를 분석하는 교육 연구가 다수 이루어졌으며 유의미한 결과를 도출하였다.

최근 빠르게 변화하는 기술과 사회에 적응할 수 있는 능력이 강조됨에 따라 지식을 얻는 과정이나 방법을 중시하며 이러한 배경을 중심으로 정보 교과서의 탐구성에 대한 연구가 다각도로 이루어졌다. 정량적으로 탐구성을 측정할 수 있는 분석 방법인 Romney 분석을 활용하여 중학교 정보 교과서의 영역 혹은 일부 단원에 대해 본문, 그림, 활동, 평가 등의 대상을 중심으로 탐구성 분석이 이루어졌으며 컴퓨팅 사고력 증진을 위해서 대상에 따라 균형있는 탐구성을 가지도록 구성되어야 한다는 시사점을 도출하였다[3, 6].

또한 다른 관점의 교과서 분석 연구로 학습 목표를 선정하고 진술할 때 참고하는 교육 목표를 Anderson의 교육 목표 분류법을 기준으로 분석하였으며 학생들의 컴퓨팅 사고력 증진을 위해 절차와 메타인지 같은 고위 수준의 인지 유형과 ‘개발하다’와 ‘평가하다’에 해당하는 인지과정의 비중을 높일 필요가 있음 확인하였다[4, 5].

교과서의 텍스트나 미디어 등과 같은 자료의 특성을 파악하고 총체적으로 텍스트의 맥락을 통찰하는 내용 분석법을 이용한 정보 교과서 분석 연구에서는 2015 개정 정보 교육과정에서 강조하는 컴퓨팅 사고력의 추상화에 초점을 맞추어 채택률이 높은 7개의 중학교 정보 교과서를 분석하고 추상화 개념의 사용 실태와 컴퓨팅 사고력의 주요 요소인 자동화를 염두한 추상화의 필요성 그리고 교육과정의 범위와 위치에 적절한 소재 활용이 필요함을 확인하였다[7].

또한 연구자의 주관적인 관점을 최소화하고 데이터 마이닝 기법을 활용하여 객관적으로 교과서의 내용을 분석한 연구 사례도 확인할 수 있었다. 2015 개정 교육과정과 3종의 중학교 정보 교과서의 문제해결과 프로그래밍 단원 속 단어 빈도 및 단어 간 연관 분석을 실시하여 문제, 해결, 알고리즘, 프로그래밍과 같은 핵심 개념의 제시를 확인하였다[8].

정보 교과서의 학습 문제를 주요 연구 대상으로 설정한 선행 연구는 이루어지지 않았지만 유사한 연구로 초등 실과 교과서 SW교육 단원의 학습 문제를 문제의 내용요소, 문제의 비구조성, 문제의 맥락을 중심으로 분석한 연구가 선행되었다. 특히 문제의 비구조성을 실제성, 복잡성, 개방성으로 구분하고 4점 척도로 수치화하여 분석하였으며 PISA 수학의 맥락인 개인적, 직업적, 사회적, 과학적 맥락을 중심으로 분류하여 결과를 분석하고 문제의 다양성을 확보하기 위한 전략이 필요함을 도출하였다[9].

학습 문제는 학생들에게 직접적으로 문제 해결 경험을 제공하는 주요 수단이며 수업의 설계와 실행에 질적 영향을 미치게 되고 이는 학생의 학습에도 영향을 미치게 된다[10]. 하지만 지금까지 학습 문제에 대한 체계적인 분석은 초등 실과 교과서의 SW교육 단원에 대해서만 이루어졌으며 문제의 구조성을 재해석된 척도를 활용하여 결론을 도출했다는 한계를 가지고 있음을 확인하였다[9].

2.2 문제 해결과 문제의 유형

정보 교과서의 핵심 역량인 컴퓨팅 사고력은 컴퓨팅을 기반으로 한 문제 해결력으로 요약할 수 있으며 문제 해결력을 길러주기 위해서는 학생들이 경험하게 되는 문제 자체에 대한 심도 있는 분석이 필요하다[1, 2]. 이에 문제 해결의 정의와 문제의 유형에 대한 관련 연구를 먼저 탐색하여 그 의미를 분석하였다.

문제 해결에 대한 정의는 오랜 기간 동안 여러 연구자의 관점에 따라 다양한 형태로 나타났다. Polya(1973)는 문제 해결을 ‘어려움을 극복하고, 장애물을 우회하며, 즉시 이해되지 않는 목표를 달성하는 것’으로 정의했으며 Schoenfeld(1980)는 문제는 해결 방법을 모르는 경우에만 문제라고 주장하며, 예상치 못한 놀라움이 없는 문제는 연습 문제에 불과하다고 주장하였다. Voskoglou(2012)는 문제 해결을 ‘장애물(문제)을 극복하고 우리를 둘러싼 세계에 대한 더 나은 이해를 개발하는 인지 또는 물리적 수단을 사용하는 활동’으로 정의하였다[11-13].

문제 해결에 대해 하나의 통일된 정의가 이루어지는 않았지만 많은 심리학자와 교육학자들은 문제 해결 능력이 교육의 중요한 결과 중 하나이며 교육의 목적이 사람들이 더 나은 문제 해결자가 되도록 생각하고 이성적 능력을 사용하는 것이라는 공통된 관점을 가지고 있음을 확인할 수 있다[14].

학습자의 문제 해결 능력에 대한 중요성이 강조되고 이를 길러 주기 위한 교수학습 환경 구성을 중심으로 다양한 연구가 이루어졌다. Jonassen(1997)은 교육 설계와 관련 된 연구를 통해 잘 구조화된 문제와 구조화 되지 않은 문제를 구별할 필요가 있음을 주장하며 잘 구조화된 문제에 대한 교육 설계는 정보 처리 이론에 기반하고 구조화 되지 않은 문제에 대한 교육 설계는 구성주의와 상황 인식에 기반해야 한다는 교육 설계 모델을 제시하였다[15].

또한 문제 해결의 정의가 다양한 만큼 문제 해결 경험 또한 일관되지 않으며 문제의 다양한 변인에 의해 영향을 받는다고 설명하였다. 특히 학습자가 직접적으로 경험하게 되는 문제의 유형에 초점을 맞춰 세부적으로 문제의 유형을 나누어 분석하였으며 Mayer 외(1996)에 의한 선행 연구에서 주로 문제의 구조성을 잘 구조화 된 문제와 구조화 되지 않은 문제로 나누는 이분법적인 기준을 제시하였다는 한계를 언급하였다. 이를 통해 Jonassen은 잘 구조화된(구조성이 높은) 문제에서 잘 구조화 되지 않은(구조성이 낮은) 11가지의 연속성을 가지는 문제 유형을 제안하였다[16, 17].

또한 Jonassen은 연속적인 문제 유형들은 구조화된 문제가 구조화되지 않은 문제의 전제 조건이 될 수 있으며 각 유형이 문제 해결자의 경험과 맥락에 따라 절대적이지 않음을 안내하였으며 Table 1과 같이 문제 유형에 따른 학습 활동, 투입물, 성공 기준, 맥락, 구조성, 추상성에 대한 세부 사항을 정리하였다[16].

Table 1. Detail about the Types and Structures of the Problems

Type	Learning Activity	Inputs	Success Criteria	Context	Structuredness / Abstractness
Logical Problems	logical control and manipulation of limited variables solve puzzle	puzzle	efficient manipulation number of moves or manipulations required	abstract task	discovered / abstract, discovery
Algorithmic Problems	procedural sequence of manipulations algorithmic process applied to similar sets Calculating or producing correct answer	formula or procedure	answer or product matches in values and form	abstract, formulaic	procedural predictable / abstract, procedural
Story Problems	disambiguate variables select and apply algorithm to produce correct answer using prescribed method	story with formula or procedure embedded	answer or product matches in values and form correct algorithm used	constrained to pre-defined elements, shallow context	well-defined problem classes procedural predictable / limited situation

Rule-Using Problems	procedural process constrained by rules select and apply rules to produce system-constrained answers or products	situation in constrained system finite rules	productivity or useful answers or products	purposeful academic real world constraint	unpredicted outcome / need-based
Decision making Problems	identifying benefits and limitations weighting options selection alternative and justifying	decision situation with limited alternative outcomes	answer or product matches in values and form	life decisions	finite outcomes / personally situated
Diagnosis-Solution Problems	troubleshoot system faults select and evaluate treatment options and monitor apply problem schemas	complex system with faults and numerous optional solutions	strategy used effectiveness and efficiency of treatment justification of treatment selected	real world technical mostly closed system	finite faults & outcomes / problem situated
Strategic Performance Problems	applying tactics to meet strategy in real-time, complex performance maintaining situational awareness	real-time, complex performance with competing needs	achieving strategic objective	real-time performance	ill-structured strategies well-structured tactics / contextually situated
Case Analysis Problems	solution identification, alternative actions, argue position	complex leisure-time system with multiple ill-defined goal	multiple unclear	real world, constrained	ill-structured / case situated
Design Problems	acting on goals to produce artifact problem structuring & articulation	vague goal statement with few constraints requires structuring	multiple, undefined criteria no right or wrong only better or worse	complex, real world degrees of freedom limited input & feedback	ill-structured / problem situated
Dilemmas	reconciling complex, non-predictive, vexing decision with no solution perspectives irreconcilable	situation with antinomous positions	articulated preference with some justification	topical complex, inter-disciplinary	finite outcomes, multiple reasoning / issue situated

2.3 컴퓨팅 사고력과 프로그래밍 교육

Wing(2008)의 연구로 인해 촉발된 컴퓨팅 사고력과 관련된 논의 이후 컴퓨팅 사고력의 중요성을 강조하는 관점을 넘어서 컴퓨팅 사고력 신장을 위한 다양한 방법적인 측면이 지속적으로 연구되어 왔으며 특히 프로그래밍이 컴퓨팅 사고력을 길러주기 위한 핵심적인 활동으로 강조되고 있다[18].

Voskoglou(2012)는 비판적 사고와 컴퓨팅 사고력 및 지식이 상호작용하는 문제 해결의 과정을 제시하였으며 특히 컴퓨팅 사고력은 프로그래밍 학습 전에 선행되어야 하는 기초적인 사고 과정임과 동시에 프로그래밍이 컴퓨팅 사고력을 증진 시킬 수 있는 가장 명시적인 방법임을 확인하였다[13].

같은 맥락으로 컴퓨팅 사고력과 프로그래밍 교육 연구에서 체계적인 교육을 위한 분류 체계 중심의 연구가 활발히 진행되었다. Selby(2015)는 컴퓨팅 사고력과 프로그래밍을 교육학적 관점에서 연구하였으며 산업, 학계, 교육 전문가로 구성된 구성원으로 부터

설문, 토론, 대면 인터뷰 등을 통해 얻은 데이터를 그라운드드 이론을 기반으로 질적 분석하고 컴퓨팅 사고력을 길러주기 위한 프로그래밍 교육의 순서를 Table 2와 같이 제시하였다[19].

Table 2. Order of Teaching Programming

Order	Teaching Programming
1	Constructs, Facts, Types
2	How individual constructs work
3	Use programming constructs in contrived contexts
4	Discriminate, Decompose, Abstract
5	Create programs Algorithm design
6	Test, Evaluation

프로그래밍 교육의 순서를 구조, 사실, 유형에 대한 이해부터 테스트 및 평가의 6단계로 제시하였으며 해당 순서가 모든 교육환경에 적용되는 유일한 순서는 아니지만 일반적으로 앞선 단계가 이후의 단계를 위한 필요조건이 될 수 있음을 언급하였다[19].

또한 프로그래밍 교육과 관련된 선행연구로 컴퓨팅 사고력의 핵심이 되는 자동화를 기반으로 한 문제 해결의 경험을 제공하기 위해서는 프로그래밍의 핵심 및 기초 개념을 중심으로 프로그래밍 교육 순서에 따른 순차적인 과제 제시가 필요함을 강조한 연구 결과도 확인할 수 있다[20].

선행 연구를 종합하면 컴퓨팅 사고력 향상을 위해 학생들에게 문제 해결 경험을 제공하는 정보 교과서의 학습 문제에 대한 연구가 아직 많이 부족함을 확인할 수 있다. 따라서 문제의 유형과 프로그래밍 순서 등과 같은 기준을 활용하여 학습 문제를 체계적으로 분석한다면 학습 문제에 대해 심도 있게 이해할 수 있으며 이를 통해 2022 개정 정보 교육과정에 적합한 학습 문제에 대한 시사점을 도출할 수 있을 것이다.

3. 연구 설계

3.1 연구 대상

본 연구는 정보 교과서의 체계적인 학습 문제 분석을 위해 현재 모든 학생에게 적용되는 2015 개정 정보 교육과정을 중심으로 개발된 중학교 정보 교과서의 학습 문제를 분석하고자 한다. 또한 김수환(2018)의 선행 연구에서 제시된 선택 비율이 높은 7개의 정보

교과서에서 제시된 학습 문제를 연구 대상으로 설정하였다[7].

또한 선행 연구 분석을 통해 프로그래밍이 컴퓨팅 사고력을 길러 줄 수 있는 효과적인 교육 방법 중 하나임을 확인하였다. 이를 바탕으로 정보 교과서의 전체 단원 중 컴퓨팅 활동 즉, 프로그래밍이 명시적으로 제시되며 프로그래밍을 통해 문제 해결 경험을 갖게 되는 문제해결과 프로그래밍 단원을 세부 대상으로 설정하였다[18-20].

교과서 속 분석 대상이 되는 학습 문제는 문제 해결의 정의를 제시하는 다양한 선행 연구를 종합하여 문제가 가지는 공통된 속성으로 시작 상태와 목표 상태, 목표 상태로 이동하기 위한 장애물 및 행동과 전략의 집합이라는 정의를 활용하여 선정하였으며 교과서에서 제시되는 학습 문제들 중 해당 정의에 부합하는 문제들로부터 선별하여 분석하였다[11-14].

3.2 연구 방법

Jonassen(2000)은 교육적 전략 수립에 있어 문제 유형에 따른 교수학습 전략의 중요성을 강조하였다. 이에 본 연구는 체계적인 학습 문제 분석을 위해 잘 구조화된(구조성이 높은) 문제에서 구조화 되지 않은(구조성이 낮은) 문제의 연속적인 11가지 문제 유형을 기준으로 분류하며 세부적인 기준으로 Table 1의 각 문제 유형에 따라 제시된 학습 활동, 투입물, 성공 기준, 맥락, 구조성, 추상성을 활용한다[16].

프로그래밍 활동을 포함한 학습 문제의 경우 프로그래밍 교육 순서에 따른 각 단계의 비율과 단계의 순차성을 분석하기 위해 Selby(2015)가 제시한 Table 2의 프로그래밍 교육 순서를 활용하여 각각의 학습 문제가 어떠한 단계에 해당되는지 분류하고 분석한다. 특히 하나의 학습 문제가 다양한 프로그래밍 교육 순서를 포함하는 경우에는 명시적으로 학습자가 해결해야 하는 개별적인 문제 형태로 제시하는 경우에 한하여 중복 분류하였다[19].

2022 개정 정보 교육과정의 교수학습 설계에서는 학생들의 컴퓨팅 사고력 신장을 위해 실제적인 삶의 맥락에서 컴퓨팅을 활용하여 문제를 해결하는 학습 과제를 제시하도록 안내하고 있다. 이에 현재 교과서 속 학습 문제들의 맥락을 분석하고 시사점을 도출하기 위해 컴퓨팅 사고력을 포함하는 PISA 2022 수학의 맥락을 기준으로 설정하여 개인적, 직업적, 사회적, 과학적 맥락으로 학습 문제를 분류하여 분석한다. 각

맥락에 대한 상세한 설명은 Table 3과 같다[21].

Table 3. Context in PISA 2022 Mathematics

Context	Details
Personal	Personal context category focus on activities of one's self, one's family, or one's peer group
Occupational	Occupational context category are centered on the world of work
Societal	Societal context category focus on one's community (whether local, national, or global)
Scientific	Scientific category relate to the application of mathematics to the natural world and issues and topics related to science and technology

분석 결과의 신뢰도와 객관성을 높이고자 연구자가 2주의 간격을 두고 3회 분석하여 각 기준에 따른 학습 문제 분류 초안을 도출하였으며 도출된 초안을 바탕으로 컴퓨터 교육 전공 석사 3인과 박사 1인이 함께 검토하고 분류의 결과에 대한 검토자 간의 이견이 있을 경우 검토 의견을 중심으로 협의하여 최종 분류를 완료하였다. 예를 들어 Figure 1의 학습 문제는 ‘의사결정 문제’로 연구자가 분류하였으나 4명의 검토자 중 3명의 검토자가 ‘규칙 사용 문제’로 이견을 제시하였다. 이에 문제에 대한 심도 있는 협의를 진행하였고 해당 문제는 여러 대안 중 한 가지를 선택하는 문제가 아닌 명확한 1개의 답을 다양한 방식으로 접근하는 문제로 정리되어 ‘규칙 사용 문제’ 유형으로 변경되었다[22].

Fastest routes and shortest travel times to appointments

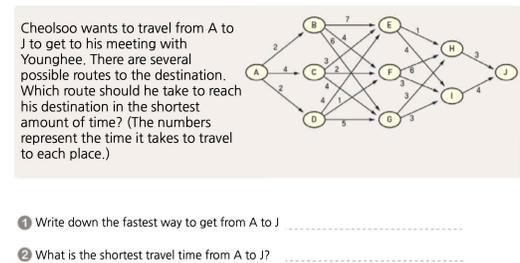


Figure 1. Examples of Learning Problems Set to A Different Type with Expert Review

4. 연구 결과 및 해석

4.1 문제 해결 정의에 따른 학습 문제 선별 결과

7종 교과서의 문제 해결과 프로그래밍 단원은 세부

적인 구성에 있어 교과서 마다 차이는 있으나 15 ~ 17 차시로 유사한 차시가 할당 되어 있으며 추상화, 알고리즘, 프로그래밍의 3개 소단원 혹은 추상화와 알고리즘을 통합하여 총 2개의 소단원으로 제시하는 형태로 구성되어 있다. 이에 공통된 기준으로 분석하기 위해 2개의 소단원을 기준으로 통합하여 분석하였다 [22-28].

교과서 내 학습 문제는 명시적으로 학생들이 특정 상태에서 직접 목표 상태로 변화 시켜야 하는 학습 문제 또는 학습 활동을 선정하였다. 명시적인 문제 상황이 주어지지 않거나 일련의 문제 해결 절차를 교과서 지문 및 자료를 통해 직접적으로 제시하는 형태의 경우 문제 해결의 정의에 따라 목표 상태로 이동하기 위한 장애물 및 행동, 전략 집합이 제시되지 않는 형태로 판단하여 제외하였다.

또한 교과서에서 객관식 및 주관식의 응답을 주로 요구하는 단원 평가는 모든 교과서에서 동일한 형태로 제시하고 있으며 같은 유형의 문제로 구별되기 때문에 비교 분석 대상에서 제외하였다. 각 교과서에서 최종적으로 연구 대상으로 선정된 학습 문제 현황은 Table 4와 같다.

Table 4. State of Learning Problems in Textbooks

Text Book	Sub-Chapter		Sum
	Abstraction and Algorithms	Programming	
A	10	11	21
B	8	21	29
C	7	16	23
D	16	17	33
E	3	20	23
F	8	17	25
G	9	13	22
Sum	61	115	176

학습 문제는 교과서 별로 차이가 있지만 21개에서 33개의 범위로 나타났으며 추상화와 알고리즘 소단원은 3개에서 16개 사이의 범위로 교과서 별로 다소 차이가 나타났다. 프로그래밍 소단원 또한 11개에서 21개의 학습 문제로 구성되어 있어 교과서 별 차이를 보였으며 모든 교과서에서 추상화와 알고리즘 소단원 보다 많은 학습 문제를 제시하였다.

4.2 학습 문제의 문제 유형에 따른 분석 결과

각 교과서의 문제해결과 프로그래밍 단원에서 제시

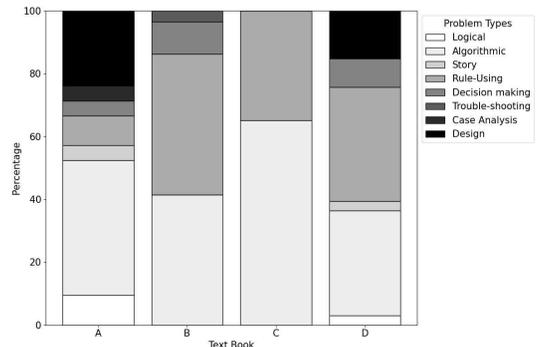
된 학습 문제를 Jonnassen의 문제 유형에 따라 분류 하였으며 분류된 학습 문제의 수와 그 비율은 Table 5와 같다.

Table 5. Analysis Results by Problem Types of Learning Problems

Textbooks	Logical	Algorithmic	Story	Rule-Using	Decision making	Trouble-shooting
A	2 (9.52)	9 (42.86)	1 (4.76)	2 (9.52)	1 (4.76)	0 (0)
B	0 (0)	12 (41.38)	0 (0)	13 (44.83)	3 (10.34)	1 (3.45)
C	0 (0)	15 (65.22)	0 (0)	8 (34.78)	0 (0)	0 (0)
D	1 (3.03)	11 (33.33)	1 (3.03)	12 (36.36)	3 (9.09)	0 (0)
E	1 (4.35)	13 (56.52)	0 (0)	6 (26.09)	0 (0)	1 (4.35)
F	2 (8.00)	5 (20.00)	0 (0)	18 (72.00)	0 (0)	0 (0)
G	0 (0)	11 (50.00)	1 (4.55)	9 (40.91)	0 (0)	1 (4.55)
Sum	6 (3.40)	76 (43.18)	3 (1.70)	68 (38.64)	7 (4.00)	3 (1.70)

Textbooks	Diagnosis-Solution	Strategic Performance	Case Analysis	Design	Dilemmas
A	0 (0)	0 (0)	1 (4.76)	5 (23.81)	0 (0)
B	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
C	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
D	0 (0)	0 (0)	0 (0)	5 (15.15)	0 (0)
E	0 (0)	0 (0)	0 (0)	2 (8.70)	0 (0)
F	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
G	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Sum	0 (0)	0 (0)	1 (0.57)	12 (6.82)	0 (0)

모든 교과서에서 알고리즘 문제와 규칙 사용 문제의 두 가지 유형 빈도가 가장 높게 나타났다. 진단-해결, 전략 수행, 딜레마 유형의 문제는 모든 교과서에서 나타나지 않았으며 이야기, 트러블 슈팅, 상황 분석, 설계 문제 유형의 경우 일부 교과서에서 소수의 빈도로 나타났다. 각 교과서의 학습 문제 구성에 대해 상세히 분석하기 위해 문제의 유형 별로 각 교과서에서 차지하는 비율을 시각화하였으며 그 결과는 Figure 2와 같다.



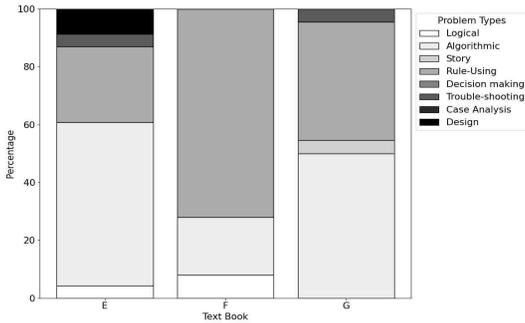


Figure 2. Visualization by Problem Types of Learning Problems

논리, 알고리즘, 이야기, 규칙 사용 문제의 유형은 한 가지 문제 해결 방법을 갖는 비교적 구조성이 높은 문제 유형에 속한다. 이를 통해 대부분의 정보 교과서는 주로 하나의 답을 갖는 구조성이 높은 문제들 위주로 구성되었다고 볼 수 있다.

이야기 문제 유형의 경우 대부분의 교과서 마지막 연습 문제에 나오는 형태인 정형화된 문제 유형으로 그 특성이 정의되지만 정보 교과서에서는 그 빈도가 낮게 나타났다. 이는 이야기 문제는 이야기 속에서 핵심 단어를 식별하고 문제를 해결하기 위한 과정의 설계가 포함되어 있는 형태이나 대부분의 정보 교과서에서 이야기를 포함하여 제시된 형태는 문제를 해결하기 위한 핵심 정보를 포함하지 않은 단순 배경 설명을 위한 목적으로만 활용된 경우가 많았다. 이러한 유형은 이야기 유형이 아닌 다른 유형으로 분류되었기 때문에 이야기 유형의 빈도가 낮게 나타났다.

Jonassen의 문제 유형은 논리 문제 유형에서 딜레마 문제 유형으로 갈수록 구조성이 낮다. A와 D 교과서의 경우 구조성이 낮은 문제 중 하나인 설계 문제의 빈도가 다른 교과서와 비교하여 상대적으로 높게 나타났다. 해당 교과서의 문제를 세부적으로 살펴보면 프로그래밍으로 문제를 해결하기 위한 기준과 제약 조건을 제시한 뒤 학생들이 자유롭게 프로그램을 개발하고 이를 통해 문제를 해결하는 경험을 제공하는 형태의 학습 문제로 일부 구성되어 있었다.

학습 문제의 유형을 분석한 결과 정보 교과서에서는 주로 구조성이 높은 학습 문제를 제시하고 있으며 알고리즘과 규칙사용 문제의 비율이 전체의 80% 이상으로 나타나 문제의 목표나 답이 명확한 형태의 학습 문제 비중이 높은 것을 확인할 수 있다.

4.3 프로그래밍 교육 순서에 따른 분류 결과

학습 문제 중 프로그래밍 활동을 포함하는 문제에 한하여 프로그래밍 교육의 순서에 따라 분류한 각 순서별 학습 문제의 수와 비율은 Table 6과 같다.

Table 6. Analysis Results by Programming Teaching Orders of Learning Problems

Step	Constructs Facts, Types	How individual constructs work	Use programming constructs in contrived contexts	Discriminate Decompose Abstract	Create programs Algorithm design	Test Evaluation
A	0 (0)	0 (0)	4 (17.39)	6 (26.09)	7 (30.43)	6 (26.09)
B	5 (20.83)	7 (29.17)	4 (16.67)	2 (8.33)	4 (16.67)	3 (8.33)
C	2 (8)	6 (24)	2 (8)	6 (24)	6 (24)	3 (12)
D	1 (4)	8 (32)	2 (8)	6 (24)	6 (24)	2 (8)
E	2 (6.25)	7 (21.88)	5 (15.63)	6 (18.75)	6 (18.75)	6 (18.75)
F	5 (23.81)	2 (9.52)	5 (23.81)	4 (19.05)	4 (19.05)	1 (4.76)
G	6 (42.86)	0 (0)	5 (35.71)	0 (0)	2 (14.29)	1 (7.14)
Sum	21 (12.72)	30 (18.18)	27 (16.36)	30 (18.18)	35(21.21)	22(13.33)

교과서에 따라 프로그래밍 교육 순서의 특정 단계를 학습 문제로 제시하지 않는 경우가 있었지만 각 순서가 비교적 고른 분포로 나타났다. 순서별 구성을 상세히 분석하기 위해 시각화한 결과는 Figure 3과 같다.

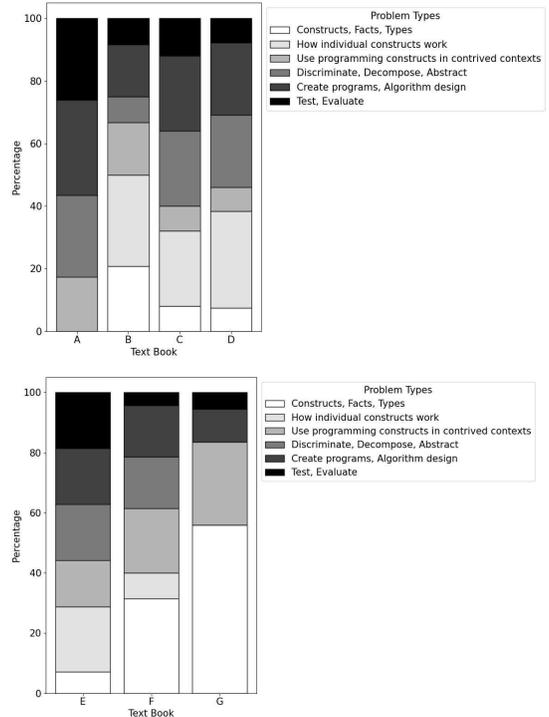


Figure 3. Visualization by Programming Teaching Orders of Learning Problems

대부분의 교과서가 6가지 프로그래밍 학습 단계 모두를 학습 문제의 형태로 구성하여 제시하고 있었으며 A와 G교과서의 경우 일부 순서가 학습 문제의 형태로 제시되지 않았다. 특히 E교과서의 경우 각 단계 별로 유사한 비율의 학습 문제를 제시하고 있었으며 A교과서는 ‘구조, 사실, 유형’의 단계에 해당하는 학습 문제를 제시하지 않았고 G교과서의 경우 프로그래밍의 ‘개별 구조 동작’에 대한 단계와 ‘구분, 분해, 추상화’와 관련된 학습 문제를 제시하지 않았다.

교과서의 구성을 상세히 살펴보면 주로 프로그래밍에 활용되는 기본 개념과 원리를 학습하는 단계가 ‘구조, 사실, 유형’, ‘개별 구조 동작’ 단계에 해당되었는데 해당 문제가 제시되지 않았던 A와 G교과서는 학습 문제를 해결하는 활동 없이 해당 개념과 원리를 직접 교수법 등을 통한 따라하기의 형태로 표현한 것을 확인할 수 있었다.

‘구분, 분해, 추상화’ 단계의 경우 알고리즘 설계 전에 문제에서 핵심 속성을 뽑아내거나 단순화 하는 과정을 주로 포함하고 있었으며 G교과서의 경우 해당 단계를 학습 문제로 명시적으로 제시하지 않고 특정 문제를 해결하기 위한 프로그램을 작성하는 형태로 포괄적으로 제시하고 있는 것을 확인할 수 있었다.

모든 교과서에서 ‘테스트와 평가’ 단계는 주로 ‘프로그래밍, 알고리즘 설계’의 단계에 이어서 제시되었으며 개별 문제의 형태로 제시되는 경우는 매우 드물게 나타났다. 또한 ‘구분, 분해, 추상화’ 단계에서 ‘테스트 및 평가’에 이르는 3가지 단계는 주로 하나의 학습 문제에서 하위 학습 문제로 나누어 제시되는 형태가 많았다. 이는 중복으로 각각의 단계에 분류되었고 학습 문제의 구성에 따라 일부 단계가 생략되는 형태로 다양하게 제시되었다.

전체적인 정보 교과서의 프로그래밍 학습 문제는 각 단계가 고르게 분포되어 있었으며 비교적 유사한 문제의 수로 구성되었고 순차적으로 제시됨을 확인할 수 있다.

4.4 학습 문제의 맥락에 따른 분석 결과

각 교과서 학습 문제의 맥락을 체계적으로 분석하기 위해 컴퓨팅 사고력을 포함하는 PISA 2022 수학의 맥락을 중심으로 학습 문제를 분류하였다. 분류된 각 맥락의 학습 문제 수와 비율은 Table 7과 같다

Table 7. Analysis Results by Context of Learning Problems

Context	Personal	Occupational	Sciatal	Scientific	Uncategorized
A	5 (23.81)	0 (0)	3 (14.29)	3 (14.29)	10 (47.62)
B	12 (41.38)	0 (0)	1 (3.45)	0 (0)	16 (55.17)
C	3 (13.04)	0 (0)	2 (8.70)	1 (4.35)	17 (73.91)
D	6 (18.18)	0 (0)	0 (0)	2 (6.06)	25 (75.76)
E	4 (17.39)	0 (0)	0 (0)	4 (17.39)	15 (65.22)
F	6 (24.00)	0 (0)	0 (0)	9 (36.00)	10 (40.00)
G	13 (59.09)	0 (0)	0 (0)	0 (0)	9 (40.91)
Sum	49 (27.84)	0 (0)	6 (3.40)	19 (10.80)	102 (57.95)

분석 결과 맥락을 분류할 수 없는 학습 문제의 비중이 가장 크게 나타났으며 개인적 맥락, 과학적 맥락, 사회적 맥락 순으로 나타났다. 모든 교과서에서 직업적 맥락과 관련된 학습 문제는 나타나지 않았다. 상세한 학습 문제의 구성을 분석하기 위해 학습 문제의 맥락 비율을 Figure 4와 같이 시각화 하였다.

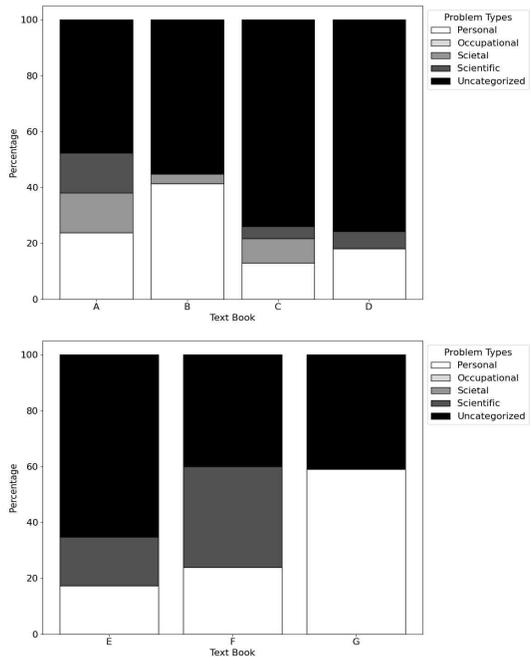


Figure 4. Visualization by Context of Learning Problems

모든 교과서에서 40% ~ 75% 이상의 학습 문제가 맥락을 분류할 수 없는 것으로 나타나 가장 큰 비중을 차지하고 있는 것을 확인할 수 있다. 그 외에 높은 비중을 차지한 맥락으로는 개인적인 맥락의 학습 문제를 확인할 수 있었다. 특히 G교과서의 경우 맥락을 분류할 수 있는 모든 학습 문제가 개인적 맥락으로 분

류되었으며 약 60%에 가까운 학습 문제가 개인적인 맥락으로 나타났다. 이를 통해 G교과서가 학생의 삶과 가장 가까운 학습 문제로 구성되었다고 해석할 수 있었다.

분류 될 수 없는 맥락의 학습 문제 비중이 높은 이유를 분석하면 정보 교과서의 문제 해결과 프로그래밍 단원에서 제시되고 있는 학습 문제는 대부분 프로그래밍을 전제로 한 학습 문제로, 프로그래밍 도구 표현의 특성으로 인해 학생들의 삶에 가까운 맥락을 효과적으로 연관 지어 문제 형태로 제시하는데 한계가 있었던 것으로 해석된다.

5. 결론 및 제언

2015 개정 교육과정과 2022 개정 교육과정에 따르면 정보 교과는 컴퓨팅 사고력을 핵심 역량으로 강조하고 컴퓨팅 사고력을 바탕으로 실생활 및 다양한 학문 분야의 문제를 해결하는 능력을 기르는 교과로 정의된다. 컴퓨팅 사고력의 정의는 다양하지만 공통적으로 컴퓨팅을 활용한 문제 해결에 초점을 맞추고 있으며 이러한 역량을 길러주기 위해서는 프로그래밍을 통한 문제 해결 경험이 교수학습 과정에서 중요한 요소로 작용하게 된다[1, 2, 18, 19, 20].

이에 본 연구는 문제와 문제 해결과 관련된 선행 연구를 분석하여 그 의미를 정리하고 교과서에서 제시된 학습 문제 분석의 필요성을 확인하였다. 체계적인 학습 문제 분석을 위해 문제의 유형, 프로그래밍 교육 순서, 맥락을 중심으로 각 교과서의 학습 문제를 분석하였으며 다음과 같은 결론을 도출하였다.

문제의 유형에 따른 분석 결과 정보 교과서의 학습 문제들은 주로 구조화 된(구조성이 높은) 형태의 문제가 높은 비중으로 나타났으며 주로 하나의 답을 가지고 있고 구조화의 정도에 따라 해결 방법까지 한정적인 문제들이 많이 제시되었다. Jonassen은 구조화된 문제와 구조화 되지 않은 문제의 차이점을 언급하며 구조화된 문제는 주로 정보 처리 이론에 기반을 두고 구조화되지 않은 문제는 구성주의에 기반하여 교육 설계가 이루어져야 함을 강조하였다. 이에 교과서 구성 시 학생들의 문제 해결 과정을 효과적으로 지원하기 위해 각각의 문제 유형에 따라 다른 관점의 교수 학습 설계가 이루어질 필요가 있다[15].

또한 구조화되지 않은 문제는 학생들의 유연한 사고를 촉진하며 지식을 구축하고 스스로 미래의 학습

을 위한 준비를 지원하는데 도움을 준다[29-31]. 이에 학생들의 문제 해결력을 길러주기 위해 구조화되지 않은 문제 해결 경험을 포함하여 다양한 학습 문제를 제공할 필요가 있다. 특히 인공지능과 관련된 다양한 문제 해결 경험의 요구가 커짐에 따라 데이터를 기반으로 귀납적으로 판단하며 학생들이 스스로 가설을 세우고 이를 바탕으로 문제를 해결하며 추측 및 논쟁을 촉진할 수 있는 문제를 제공할 필요가 있다[29-31].

문제의 유형에 따라 학습 문제의 순서를 구성하는 방법에 대한 고려도 필요하다. 구조화된 문제는 구조화되지 않은 문제를 해결하는데 전제 조건이 될 수 있으며 문제의 표현 방법에 따라 구조화된 정도를 다양하게 나타낼 수 있다. 이에 교과서에서 구조화 되지 않은 문제를 제시 할 때에는 이전 형태의 문제 유형에 대한 충분한 경험 제공이 필수적으로 선행되는 순서의 설계가 고려될 필요가 있다[16].

정보 교과서의 프로그래밍 학습 문제를 프로그래밍 교육의 순서를 중심으로 분석한 결과 모든 교과서에서 다양한 순서의 학습 문제가 비교적 고르게 순차적으로 제시되고 있음을 확인할 수 있었다. 선행 연구를 통해 교수학습 환경에서의 프로그래밍 교육은 학생들에게 단계적으로 제시되어야 하며 핵심 개념과 기초에 집중한 순차적인 과제가 필요함을 도출하였고 프로그래밍 단계에 따른 정보 교과서의 학습 문제 구성이 적절한 것을 확인할 수 있었다[19, 20].

프로그래밍의 개념을 학습하고 주어진 맥락 안에서 활용하는 단계에서 교사의 지시나 교과서의 안내에 따라 프로그래밍 하는 활동을 할 때 적합한 프로그래밍 학습 문제를 함께 제공한다면 문제 해결 경험을 통해 학습한 개념을 내제화 할 수 있어 학생들에게 더 유의미한 교수학습 환경을 제공해 줄 수 있을 것이다. 이에 프로그래밍의 구조, 개념 등을 학습할 수 있는 더 다양한 형태의 학습 문제와 전략들이 개발되고 적용될 필요가 있다[20].

학습 문제를 맥락 측면에서 분석한 결과 프로그래밍 도구 및 활동 방법에 따른 표현의 한계로 PISA 2022 수학 맥락을 기준으로 분류되지 않는 학습 문제의 비중이 가장 높게 나타났다. 그 외 일부 학습 문제가 주로 개인적 맥락과 관련지어 제시되었고 직업적 맥락은 제시되지 않았으며 소수의 사회적 맥락, 과학적 맥락의 문제를 확인할 수 있었다. 2022 개정 정보 교육과정의 교수학습 및 평가에서는 실제적인 삶의 맥락에서 컴퓨팅을 통해 문제를 해결하는 학습 문제를 제시하도록 명시하고 있다[1]. 좋은 학습 문제의

특성과 관련된 선행 연구에서도 학생들의 경험과의 관련성을 중요시 하였으며 이는 본질적인 동기를 지원하여 문제 해결력을 기르는데 효과적임을 강조하였다[29, 30]. 이에 개인적인 맥락의 학습 문제 비중이 높은 것은 올바른 학습 문제의 구성으로 판단되며 컴퓨팅 사고력을 통한 문제 해결이 삶의 맥락에서 이루어질 수 있도록 개인적인 맥락의 학습 문제가 더 많이 개발되고 적용되어야 한다.

또한 2022 개정 교육과정에서는 자신의 삶과 진로를 스스로 설계하는 자기 관리 역량과 지속 가능한 인류 공동체 발전에 기여 할 수 있는 공동체 역량을 강조하고 있다. 이에 새롭게 개발될 2022 개정 교육과정을 기반으로 한 교과서에서는 직업적, 사회적, 과학적 맥락을 포함하여 다양한 맥락이 드러나는 학습 문제 제시가 필요하다. 이러한 다양한 맥락의 문제 해결 경험을 통해 학생들은 지식을 문제 해결의 도구로 보게 될 것이며 의사소통 기술을 촉진시킬 수 있을 것이다 [29, 31].

본 연구는 정보 교과서의 학습 문제를 심층적으로 분석하고 학생들의 컴퓨팅 사고력을 길러주기 위한 학습 문제의 방향과 구성을 제안하였다. 본 연구의 결과가 2022 개정 교육과정을 비롯하여 정보 교육에 활용되는 학습 문제를 연구하는데 기초연구로 활용되어 교육현장에 도움을 줄 수 있기를 기대한다.

참고문헌

- [1] National Curriculum Information Center. (2023). *2022 revised curriculum*. National Curriculum Information Center
- [2] National curriculum information center (2023). *2015 revised curriculum*. National Curriculum Information Center
- [3] Yu, B., Kim, J., & Lee, W. (2016). Implication for construction computing system unit of the 2015 revised curriculum. *The Journal of Korean Association of Computer Education*, 19(2), 31-40.
- [4] Choe, H. (2014). Study of Analysis about Learning Objectives of Informatics Textbooks in Middle School using Anderson's Taxonomy of Educational Objectives. *The Journal of Korean Association of Computer Education*, 17(1), 51-63.
- [5] KANG, O. (2019). Analysis of the Organization System and Learning Objectives of Middle School Informatics Textbooks. *The Journal of Korean Association of Computer Education*, 22(2), 1-9. DOI: 10.32431/KACE.2019.22.2.001
- [6] KANG, O. (2019). The Analysis of Inquisitive Tendency in Middle School Informatics Textbooks Based on the 2015 Revised National Curriculum. *The Journal of Korean Association of Computer Education*, 22(2), 21-28. DOI: 10.32431/KACE.2019.22.2.003
- [7] Kim, S. (2018). Analysis of Abstraction Contents in Informatics Textbooks of Middle School According to 2015 Revised Curriculum. *The Journal of Korean Association of Computer Education*, 21(5), 1-10. DOI: 10.32431/KACE.2018.21.5.001
- [8] Choe, H. (2020). Analysis of Core Concepts in Problem Solving and Programming Unit of Informatics Subject Textbooks in Middle School Revised in 2015. *Journal of Digital Contents Society*, 21(1), 63-70. DOI: 10.9728/dcs.2020.21.1.63
- [9] Lim, J., Park, J., Kim, Y., & Lim, K. (2021). An Analysis of the Learning Problems for SW Education in Elementary Practical Arts Textbooks of the 2015 Revised Curriculum. *The Journal of Korean Association of Computer Education*, 24(2), 1-14. DOI: 10.32431/KACE.2021.24.2.001
- [10] Yeo, S., Kim, G., & Kim, D. (2023). Exploring Student Learning Opportunities with Programming Tasks in Middle School Computer Science Textbooks. *Korean Association for Educational Information and Media*, 29(1), 1-30. DOI: 10.15833/KAFEIAM.29.1.001
- [11] Polya, G. (2004). *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press.
- [12] Schoenfeld, A. H. (1980). Teaching Problem-Solving Skills. *The American Mathematical Monthly*, 87(10), 794-805. DOI: 10.1080/00029890.1980.11995155
- [13] Voskoglou, M. G., & Buckley, S. (2012). *Problem Solving and Computers in a Learning Environment*.
- [14] Gagné, R. M., & Robert M. (1985). *The conditions of learning and theory of instruction (4th ed)*. New York: Holt, Rinehart and Winston.
- [15] Jonassen, D. H. (1997). Instructional design models for well-structured and III-structured problem-solving learning outcomes. *Educational Technology Research and Development*, 45(1), 65-94. DOI:

- 10.1007/BF02299613
- [16] Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Development*, 48(4), 63-85. DOI: 10.1007/BF02300500
- [17] Mayer, R.E., & Wittrock, M.C. (1996) Problem-solving transfer. In D.C. Berliner & R.C. Calfee (Eds.), *Handbook of educational psychology* (pp. 47-62). New York: Macmillan.
- [18] Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. DOI: 10.1098/rsta.2008.0118
- [19] Selby, C. C. (2015). Relationships: Computational thinking, pedagogy of programming, and Bloom's Taxonomy. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 80-87. London United Kingdom: ACM. DOI: 10.1145/2818314.2818315
- [20] Stephenson, C., Cooper, S., Owens, B., & Gal-Ezer, J. (2012). The new CSTA K-12 computer science standards. *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE '12*. DOI: 10.1145/2325296.2325380
- [21] OECD (2023. 11. 15) Retrieved from PISA 2022 mathematics framework, <https://pisa2022-maths.oecd.org/>
- [22] Lee, W., Kim, J., Yoo, S., Kim, M., Shin, E., Kim, T. (2019). Middle School Informatics TextBook. MiraeN.
- [23] Kim, Y., Lim, J., Kim, M., Park, J., Jung, Y., Lee, S., Middle School Informatics TextBook. Geumsung.
- [24] Jung, Y., Kang, S., Kim, M., Seo, M., Son, Y., Song, S., Song, J., Shin, I., Youn, S., Lee, D., Lee, C., Lim, S., Lim, J. (2019). Middle School Informatics TextBook. Gmass.
- [25] Kim, H., Lee, Y., Kim, S., Lee, H., Kim, S. (2019). Middle School Informatics TextBook. Chunjae.
- [26] Han, S., Ryu, M., Jeon, S., Hwang, M., Kim, H., Hong, S., Seo, J. (2019). Middle School Informatics TextBook. Sungandang.
- [27] Lim, H., Kim, H., Seo, J., Kim, J., Jo, J. (2019). Middle School Informatics TextBook. Visang.
- [28] Lee, Y., Yoo, H., Choi, J., Ahn, S. (2019). Middle School Informatics TextBook. Kyohaksa.
- [29] Barrows, H., and Kelson, A. C. (1995). Problem-Based Learning in Secondary Education and the Problem-Based Learning Institute (Monograph 1), *Problem-Based Learning Institute*, Springfield, IL.
- [30] Kolodner, J. L., Hmelo, C. E., and Narayanan, N. H. (1996). Problem-based learning meets case-based reasoning. In Edelson, D. C., and Domeshek, E. A. (eds.), *Proceedings of ICLS 96, AACE, Charlottesville, VA*, pp. 188-195.
- [31] Gallagher, S. A., Stepien, W. J., and Rosenthal, H. (1992). The effects of problem-based learning on problem solving. *Gifted Child Q.* 36:195-200



김슬기

2008년 경인교육대학교
초등교육학과(교육학석사)
2016년 경인교육대학교
융합교육학과(교육학석사)
2023년 한국교원대학교
컴퓨터교육과(교육학박사)

2008년 ~ 현재 경기도 교육청 초등교사
관심분야: 정보(SW·AI) 교육, 프로그래밍 교육, 컴퓨팅사고력,
데이터 과학, 데이터 리터러시
E-Mail: tmfriska85@gmail.com

