



# 교육용 프로그래밍 언어(EPL) 기반 학습에서 문제 유형에 따른 학습 데이터 분석: 네이버 엔트리(Entry)를 중심으로\*

## Analysis of Learning Data by Problem Type in Educational Programming Language(EPL) Based Learning: Focusing on Entry Which was Developed by Naver

이지원<sup>†</sup> · 손예지<sup>††</sup> · 이명화<sup>†††</sup> · 이정민<sup>††††</sup>  
Jiwon Lee<sup>†</sup> · Yeji Son<sup>††</sup> · Myunghwa Lee<sup>†††</sup> · Jeongmin Lee<sup>††††</sup>

### 요약

본 연구는 효과적인 EPL 기반 학습 설계를 위한 기초자료를 제공하고자 Jonassen의 이론에서 제안된 열한 가지의 문제 유형에 따라 엔트리의 '발견하기' 학습 콘텐츠를 유형화하고 그에 관한 로그 데이터 총 116,927개를 분석하였다. 그 결과, EPL 기반 학습에 주로 속하는 문제 유형을 문제의 구조성 정도에 따라 분류하고, 학습 카테고리 관점과 난이도 관점에서 학습 설계를 위한 시사점을 도출할 수 있었다. 주된 시사점으로, EPL 기반 학습에서는 구조화와 비구조화 문제 특성을 동시에 고려하여 학습 과정을 설계할 필요가 있다. 그리고 학습 문제의 특성을 고려하여 적합한 문제 난이도와 문제 주제를 설계할 필요가 있다. 이를 통해 본 연구는 실제 로그 데이터 분석 결과를 기반으로 소프트웨어 교육에서 효과적인 교육용 프로그래밍 언어 기반 학습 설계를 위한 시사점을 제안하였다는 점에서 의의가 있다.

**주제어** 소프트웨어 교육, 교육용 프로그래밍 언어, 엔트리, 문제 유형, 로그 데이터 분석

### ABSTRACT

To provide a basis for effective EPL learning design, this study categorized the “discover” learning content of entries according to the eleven problem types proposed in Jonassen's theory and analyzed a total of 116,927 log data. As a result, the problem types that primarily belong to EPL-based learning were categorized according to the degree of structuredness of the problems, and implications for learning design were drawn from the perspective of learning categories and difficulty level. The main implication is that in EPL-based learning, designing a learning process that considers structured and unstructured problem characteristics is necessary. In addition, it is essential to create appropriate problem difficulty and problem topics that consider the characteristics of learning problems. The significance of this study is that it proposed implications for designing effective educational programming language-based learning based on the results of accurate log data analysis in the fields of SW education.

**Keywords** SW Education, Educational Programming Language, Entry, Log Data Analysis

†정회원 이화여자대학교 대학원 교육공학과 석사  
††정회원 이화여자대학교 대학원 교육공학과 석사  
†††정회원 이화여자대학교 대학원 교육공학과 박사  
수료  
††††정회원 이화여자대학교 대학원 교육공학과 교수  
(교신저자)  
논문투고 2024년 10월 28일  
심사완료 2025년 02월 18일  
게재확정 2025년 02월 19일  
발행일자 2025년 03월 12일

\* 본 논문은 네이버 커넥트 연구재단의 지원을 받아 수행된 연구임

## 1. 서론

디지털 대전환 시대에 살아가는 학생들은 웹, 모바일 기기를 통해 과도한 지식과 혼재된 정보를 습득하고 있다. 일상에서 방대하고 모호한 정보들을 처리할 능력이 없어 받아들이기만 한다면, 삶에서 중요한 가치인 문제를 인식하고, 해결하는 방향을 확립해가기 어려워진다. 그로 인해 최근 교육에서는 학생의 컴퓨팅 능력으로 삶에서 마주하는 복잡한 문제를 다양한 지식을 융합하여 해결하는 학습자의 주체적인 역량이 강조되고 있다[1-2]. 이는 2022 개정 교육과정 총론에 근거하며 이러한 가치를 실천하기 위해 초등학교 때부터 놀이 중심의 프로그래밍을 통해 디지털 역량을 향상하는 과목이 신설될 전망이다[1].

한편, 프로그래밍 교육에 학습자가 즐겁게 참여하기 위해서 엔트리, 스크래치 같은 교육용 프로그래밍 언어(Educational Programming Language, 이하 EPL) 플랫폼이 교육 현장에서 사용되고 있다[3]. 학교에서 EPL 플랫폼을 사용할 때는 주로 문제 기반 학습전략을 활용하는데, 이는 학습자의 흥미로운 문제 풀이 경험을 촉진하여 문제해결력을 향상할 뿐만 아니라 프로그래밍적 문제해결력인 컴퓨팅 사고력을 향상하려는 목적이 있다[4-6].

EPL 문제기반학습을 위해서는 학습자에게 제시되는 문제를 체계적으로 설계하는 것이 매우 중요한데, 문제의 유형에 따라 문제에 대한 정의방법, 해결방안을 위한 경로 등 문제 해결과정이 달라지기 때문이다[7-8]. Alshaye 외[9]에 따르면, 문제기반 학습에서 학습자들에게 제공되는 문제는 학습자들의 문제해결 역량과 프로그래밍 역량에도 큰 영향력을 미치는 것이 확인되었다. 즉, EPL 환경에서 문제 기반 학습 효과를 높이기 위해서는 난이도 및 흥미 등을 고려해 체계적으로 설계된 문제를 제공할 필요가 있음을 시사한다[10].

따라서 본 연구에서는 학습자들의 문제해결 과정을 EPL 기반 학습 환경에서 수집한 로그데이터를 활용하여 탐색하고자 하였다. Jonassen[11]의 문제 유형에 대한 이론을 기반으로 학습자가 문제해결 과정에서 마주하는 문제를 체계적으로 분석하고, 문제 유형별 학습자들의 문제해결과정을 분석하고자 하였다. 이를 통해 본 연구에서는 EPL 기반 학습에서 문제해결력 향상을 위한 효과적인 학습 설계의 시사점을 제안하는데 연구의 목적이 있다.

본 연구에서 문제를 유형별로 분석하고자 하는 목적은, 학습에서 새로운 지식을 구성하는 방식이 학습 상황에서 학습자가 어떤 문제를 마주하는지에 따라 좌우되기 때문이다[12]. 즉, 학습에서 마주치는 문제의 특성에 따라 학습자의 문제해결력 같은 학습을 통해 축적된 역량 향상에도 변화가 있을 수 있다는 것을 의미한다. 만일, 특정 학습 환경 속에 놓인 학습자가 구조화(well-structured)된 문제에 익숙해져 있다면, 실제 삶에서 인간이 마주하는 복잡한 성격의 문제를 해결하는데 필요한 지식 및 기능이 도태되어 있을 수 있다[11].

그러므로, Jonassen[11]의 문제 해결의 설계 이론은 문제 기반 학습에서 학습자 개인의 수준에 맞게 난이도를

고려한 다양한 영역의 문제들을 학습자에게 제시해 주어야 한다고 주장하며, 논리(logical), 알고리즘(algorithmic), 이야기(story), 규칙 사용(rule-using), 의사결정(decision making), 트러블 슈팅(trouble-shooting), 진단-해결(diagnosis-solution), 전략적 수행(strategic performance), 사례 분석(case analysis), 설계(design) 그리고 딜레마(dilemmas)의 총 열한 가지 문제 유형을 제안하였다. 이러한 문제 유형은 주로 목표가 분명하고 주어진 정보만으로도 문제 해결이 가능한 구조적 문제와 문제가 명확하게 제시되어 있지 않고, 제시된 정보만으로는 문제 해결이 어렵거나 불가능한 비구조적인 문제로 구분된다.

이에 따라 본 연구는 Jonassen[11]의 문제 해결의 설계 이론에서 제안된 문제 유형에 따라 EPL 기반 학습에서 학습자가 직면하는 문제를 유형화하고, EPL 기반 학습에서 수집한 로그 데이터를 분석하여 효과적인 EPL 기반 학습 설계를 위한 기초자료를 제공하고자 한다. 이를 위한 본 연구의 문제는 다음과 같다.

연구문제 1. EPL 기반 학습에서 학습자의 문제 해결 유형은 어떠한가?

연구문제 2. EPL 기반 학습에서 학습 데이터 분석 결과는 어떠한가?

연구문제 2-1. 문제 해결 주제 카테고리별 학습 데이터 분석 결과는 어떠한가?

연구문제 2-2. 문제 해결 난이도별 학습 데이터 분석 결과는 어떠한가?

## 2. 이론적 배경

### 2.1 EPL에서의 문제 기반 학습

EPL은 교육을 목적으로 개발되어 학습자의 컴퓨팅 사고, 창의적 문제해결력 및 사고력을 기르기 위해 만들어진 교육용 프로그래밍 언어를 의미한다[5]. 이러한 EPL을 통해 전문가가 아닌 초보 단계의 학습자들은 프로그래밍 언어에 대해 쉽고 흥미롭게 학습할 수 있다[3]. EPL의 대표적인 예시로 스크래치(Scratch), 엔트리(Entry) 등이 있다.

EPL 기반 학습은 스토리텔링, PBL 등의 학생참여 중심의 수업 방법을 적용하여 학습자들의 참여를 이끌고, 학습자들이 흥미를 느낄 수 있도록 한다[13-14]. 특히, 학습자들에게 문제를 제시한 뒤 해결방안을 탐색하도록 하는 문제기반학습(Problem Based Learning, 이하 PBL)은 학습자들에게 실재감을 가지고 학습에 몰입할 수 있도록 하며, 학습자들이 학습에 흥미를 느낄 수 있도록 한다는 점에서 EPL 기반 학습에 적합한 교수 방법임이 확인되었다[15].

PBL을 적용한 EPL 기반 학습 관련 선행연구 분석 결과, 이영석[4]은 대학생을 대상으로 파이썬 혹은 스크래치를 활용하여 성적 처리 프로그램 만들기 등의 활동을 통해 학습자의 전공과 관련된 실생활 문제를 해결하도록 하였는데, 그 결과 학생 수준에 적합하고 실생활에 관련된 문제일수

록 학생의 흥미와 컴퓨팅 사고력 향상에 도움을 주는 것으로 나타났다. 또한, 한영신[6]은 문제를 학습 주제와 관련된 범위 내에서 자율적으로 탐색하도록 한 후 이를 스크래치를 활용하여 해결할 수 있도록 하여 EPL 기반 학습에서 마주할 수 있는 실질적 문제해결을 수행하도록 하여 학습자들의 컴퓨팅 사고력과 문제해결력 향상에 효과적임을 밝혔다. 아울러 비전공자를 대상으로 문제 기반 학습을 적용한 고광일[16]에 따르면, Scratch와 같은 EPL 학습 환경에서 실제적인 문제를 해결한 실험집단이 문제 기반 학습을 적용하지 않은 집단보다 학습 효과 및 컴퓨팅 사고력 향상에 긍정적인 영향을 미친 것으로 나타났다. 마지막으로 엔지니어링 디자인과 컴퓨터 과학과 관련된 예술, 스토리텔링, 게임의 문제를 제공한 Strawhacker와 Bers[17]의 연구결과에 따르면, PBL을 적용한 EPL 기반 학습은 학습자들의 인지 발달에 긍정적 영향을 미쳤다고 보고하였다.

이와 같은 선행연구를 종합해 보면, PBL을 적용한 EPL 교육은 어린 학습자부터 대학생까지 다양한 연령에서 효과적인 교수 방법이며, 이는 학습자들에게 문제를 제공해 주고 이에 대한 해결책을 도출하도록 한다는 점에서 학생들에게 적절한 문제를 제공해야 함이 확인되었다.

## 2.2 프로그래밍 문제 유형 분류 기준

우리가 학습에서 마주하는 문제의 속성은 인지적, 정의적, 귀납적으로 다르므로, 각 문제의 특성에 따라 서로 다른 교수적 처방이 필요하다[11]. 문제해결력 향상을 위한 프로그래밍 학습에서도 효과적인 학습 설계를 위해 학습 문제 자체에 미치는 다양한 변인을 고려하여 그 특성을 고찰하는 과정이 중시되고 있다[18]. 이에 따라 본 연구는 EPL 기반 학습 환경에서 각 문제 특성을 고려한 학습 설계 방안을 시사하기 위해 Jonassen[11]이 제시한 문제 유형

을 바탕으로 EPL 환경의 학습 문제별 특성에 따라 유형화하고자 하였다. Jonassen이 제안한 문제 유형의 경우 광범위한 문제들을 기반으로 도출되어 분류체계가 타당화되어 있으며, 문제의 다양한 특성을 반영할 뿐 아니라 문제 유형별 맞춤형 교수적 지원의 필요성을 강조하기 때문에 본 연구에 적절하다고 판단되어 이를 활용하였다.

그의 문제 설계 이론에 따르면, 문제 유형은 논리(logical), 알고리즘(algorithmic), 이야기(story), 규칙 사용(rule-using), 의사결정(decision making), 트러블 슈팅(trouble-shooting), 진단-해결(diagnosis-solution), 전략적 수행(strategic performance), 사례 분석(case analysis), 설계(design) 및 딜레마(dilemmas)의 총 열한 가지로 구성되어 있다. 그리고 이는 학습 활동(learning activity), 투입물(inputs), 문제 해결 기준(success criteria), 문제 맥락(context), 구조도(structuredness), 추상도(abstractness)의 6가지 기준에 따라 분류된다. 이 중에서 특히, 문제의 구조화 정도는 구조도와 추상도를 토대로 분류될 수 있고, PBL에서 학습자의 성과는 구조화와 비구조화 문제 유형에 따라 달라질 수 있으므로 학습 설계 자체도 차별화될 수 있다[19].

이에 따라 본 연구는 Jonassen[11]의 열한 가지 문제 유형을 구조화 정도에 따라 분류하였다. 본 연구에서는 프로그래밍 문제 특성을 고려한 선행연구에 따라 EPL 기반 학습에서 구조화 정도에 가까운 문제와 비구조화 정도에 가까운 문제를 분류하였다[18]. 먼저, 구조화 문제(well-structured problems)는 제한된 수의 규칙이나 원칙을 기반으로 잘 정의된 매개 변수(parameters)를 활용하여 수렴적인 해결책을 탐색하는 어느 정도 제약(constrain)이 있는 문제를 의미한다[19]. 구조화 문제에 가까운 문제 유형은 Table 1과 같이 분류될 수 있다.

Table 1. Classification Criteria of Structured Problem

	Logical Problems	Algorithmic Problems	Story Problems	Rule-Using Problems
Learning Activity	logical control and manipulation of limited variables; solve puzzle	procedural sequence of manipulations; algorithmic process applied to similar sets of variables; calculating producing correct answer	disambiguate variables; select and apply algorithm to produce correct answer using prescribed method	procedural process constrained by rules; select and apply rules to produce system-constrained answers or products
Inputs	puzzle	formula or procedure	story with formula or procedure embedded	situation in constrained system; finite rules
Success Criteria	efficient manipulation; number of moves or manipulations required	answer or product matches in values and form	answer or product matches in values and form; correct algorithm used	productivity(number of relevant or useful answers or products)
Context	abstract task	abstract formulaic	constrained to pre-defined elements, shallow context	purposeful academic, real world, constrained
Structuredness	discovered	procedural predictable	well-defined problem classes; procedural predictable	unpredicted outcome
Abstractness	abstract, discovery	abstract, procedural	limited simulation	need-based

\*The contents of this table are referred to Jonassen (1997)

학습 상황에서 구조화 문제 해결을 고려한다면, 주로 정보 처리 이론과 맥을 같이 한다고 볼 수 있다. 구조화 문제는 예측할 수 있고 원칙적인 방안으로 제한된 규칙을 활용하여 해결할 수 있으며, 규칙적인 특성을 보이고, 정확하고 규정된 해결 과정이 있다는 특징이 있다. 선행연구에서 활용된 구조화 문제 분석 결과, 3, 6, 9 게임 규칙을 활용한 게임 만들기[20], 계산기 작동 원리를 반영한 계산기 만들기[4], 최대공약수와 최소 공배수를 활용한 수학 문제[21]가 있다.

EPL 기반 학습에서 구조화 문제를 활용한 선행연구에 관해, 프로그래밍 교육의 구조화 문제 유형에 대하여 시사한 김슬기[18]는 중학교 정보 교과서에서 다루는 문제의 특성을 Jonassen[11]의 분류 기준에 따라 분석하였다. 그 결과, 현재 국내 정보 혹은 SW교육에서 다루는 문제는 주로 한 가지의 답을 갖는 구조성이 높은 경우가 주를 이루고 있으며, 특히 알고리즘(algorithmic) 문제와 규칙 활용(rule-using) 문제가 전체 비율 중에서 80%를 차지하며 가장 높게 다뤄지고 있었다. 반면, 의사결정(decision

making), 트러블 슈팅(trouble-shooting), 진단 해결(diagnosis-solution), 전략 수행(Strategic 수행), 사례 분석(case analysis), 설계(design), 딜레마(dilemmas)의 비구조화된 문제처럼 문제의 구조성이 점차 애매하고 복잡해지는 문제는 교과서에서 구조화 문제에 비해 상대적으로 잘 다루이지 않거나 아예 없었다고 보고하였다. 이를 통해 전반적으로 프로그래밍 교육과정에서는 비구조화 문제보다 구조화 문제가 더욱 많다는 것이 확인되었다.

다음으로, 비구조화 문제(ill-structured problems)란 해결책, 해결 방식, 조작하기 어렵고, 매개 변수가 많은 문제를 의미하며 해결을 위해 필요한 법칙, 규칙, 개념들이 무엇인지가 불확실한 문제를 의미한다[19]. 비구조화 문제에 가까운 문제 유형은 Table 2와 같이 분류될 수 있다.

비구조화 문제 해결에서는 문제 해결을 위한 새로운 이론, 구성주의 방식의 학습, 상황 인지 접근 방식이 고려된다. 이러한 문제는 모호하게 정의되어 있거나 불분명한 목표를 가지며, 조작 가능한 매개 변수가 적고, 여러 가지 해결책이 없다. 특히, 학습자가 문제를 정의할 때 직접 판단

Table 2. Classification Criteria of Unstructured Problem

	Decision making Problem	Trouble-shooting Problems	Diagnosis-Solution Problems	Strategic Performance Problems	Case Analysis Problems	Design Problems	Dilemmas
Learning Activity	identifying benefits and limitations; weighting options; selecting alternative and justifying	examine system; run tests; evaluate results; hypothesize and confirm faults using strategies (re-place, serial elimination, space split)	troubleshoot system faults; select and evaluate treatment options and monitor; apply problem schemas	applying tactics to meet strategy in real-time, complex performance maintaining situational awareness	solution identification, alternative actions, argue position	acting on goals to produce artifact; problem structuring & articulation	reconciling complex, non-predictive, vexing decision with no solution; perspectives irreconcilable
Inputs	decision situation with limited alternative outcomes	malfunctioning system with one or more faults	complex system with faults and numerous optional solutions	real-time, complex performance with competing needs	complex, leisure time system with multiple ill-defined goals	vague goal statement with few constraints; requires structuring	situation with antinomous positions
Success Criteria	answer or product matches in values and form	fault(s) identification; efficiency of fault isolation;	strategy used; effectiveness and efficiency of treatment; justification of treatment selected	achieving strategic objective	multiple, unclear	multiple undefined criteria; no right or wrong only better or worse	articulated preference with some justification
Context	life decisions	closed system real world	real world, technical, mostly closed system	real world, performance	real world, constrained	complex, real world; degrees of freedom; limited input & feedback	topical complex, inter-disciplinary
Structuredness	finite outcomes	finite faults & outcomes	finite faults & outcomes	ill-structured strategies; well-structured tactics	ill-structured	ill-structured	finite outcomes, multiple reasoning
Abstractness	personally situated	problem situated	problem situated	contextually situated	case situated	problem situated	issue situated

\*The contents of this table are referred to Jonassen (1997)

을 내리고, 문제에 대한 개인적인 신념이나 의견을 주장하는 것이 장려된다는 특징이 있다. 선행연구에서 활용된 비구조화 문제 분석 결과, 자유롭게 프로그램을 구성하기 [20], 게임 배경, 캐릭터 등을 설정하여 직접 스토리텔링한 뒤 프로그래밍 문제를 해결하기 같은 문제[17]가 있다.

EPL 기반 학습에서 비구조화 문제를 활용한 선행연구 분석 결과, Hulls와 Rennick[22]은 대학생을 대상으로 비구조화 프로그래밍 문제에 관한 문제해결력을 향상하기 위하여 소프트웨어 플랫폼을 설계한 후 실험집단과 통제집단의 준 실험 설계를 바탕으로 교육적 효과성을 평가하였다. 통제집단은 강의 중심의 학습 활동이 주가 되었던 수업이 진행되었고, 실험집단은 비구조화 문제를 기반으로 강좌를 재설계한 수업이었다. 이때, 학생들이 해결한 비구조화 문제는 팀 기반으로 로봇 시스템에 관한 문제를 정의하고, 설계하고 구현하는 것이다. 그 결과 비구조화 문제를 중심으로 다루는 프로그래밍 학습을 성공적으로 완수한 학생들은 통제 환경에서보다 문제해결력과 자기 효능감 또한 향상되었다는 점이 확인되었다.

반면, 해당 연구에서는 비구조화 문제만을 고려하여 효과성을 검증하였으며, 구조화 문제와 비구조화 문제 간의 학습 결과가 어떠한지를 전반적으로 시사하지는 않았다. 따라서 본 연구에서는 EPL 상황에서 구조화와 비구조화 문제를 동시에 고려한 학습 설계 방향에 관해 전반적으로 시사하고자 한다.

### 2.3 EPL 기반 학습에서의 데이터 분석

EPL 기반 학습에서 학습 데이터에 관한 선행연구 분석 결과는 다음과 같다. 먼저, Jiang과 동료들[23]은 블록 기반 프로그래밍 환경에서 학습자들이 프로그래밍에서 사용한 블록 데이터를 추적하여 학습자들의 문제 해결 유형 및 문제 중심의 프로그래밍 학습 방법에 관해 시사했다. 해당 연구에서는 데이터 기반 방법(data-driven method)을 통해 학습자들의 학습 과정에서 마주한 어려움, 의미와 같은 추상적인 개념을 데이터 분석 결과에 기반하여 사실적으로 도출한 의의가 있다.

Grover과 동료들[24]의 연구에서도 블록 기반 프로그래밍 환경의 학습 데이터를 활용하여 컴퓨팅 사고력에 의미 있는 학습 분석 프레임워크를 개발하였다. 해당 연구에서는 학습자들이 남긴 로그 데이터(플레이 버튼 클릭 횟수, 플레이 버튼을 연속해서 클릭한 횟수, 파라미터를 변경한 횟수 등)를 분석함으로써 학습자들의 행동 패턴을 유추하기 위한 근거를 제공했다는 의의가 있다. 따라서 이러한 데이터 기반 연구방법을 통하여 분석하고자 하는 대상을 더욱 사실적으로 해석할 수 있고, 관련해서 수집된 근거를 바탕으로, 더욱 학습적으로 풍부한 예측이 가능하다.

이와 같은 선행연구의 결과를 바탕으로 본 연구는 EPL 중 하나인 엔트리 학습에서 제공하는 학습 콘텐츠를 문제 유형에 따라 분석하고자 한다. 엔트리는 우리나라에서 개발되어 한글 기반으로 접근성이 높고, 국내에서 가장 많이

사용되는 대표적인 EPL 플랫폼으로 알려져 있다[25-26]. 이에 따라 연구 표본의 대표성을 확보하기 위해 본 연구에서는 엔트리 학습콘텐츠에서 제공하는 학습 데이터를 분석하여 학습자들의 학습 현황과 연계하고, 이를 토대로 EPL 기반 환경에서 효과적인 학습을 설계하기 위한 시사점을 제언하고자 한다.

## 3. 연구방법

본 연구는 EPL 기반 학습에서 문제의 특징을 추출하여 Jonassen[11]의 문제 해결의 설계 이론을 토대로 문제를 유형화한 뒤, 유형화한 문제에 따라 학습자들의 학습 결과에 관한 데이터를 분석하고자 하였으며, 구체적인 연구 절차는 Fig. 1과 같다.

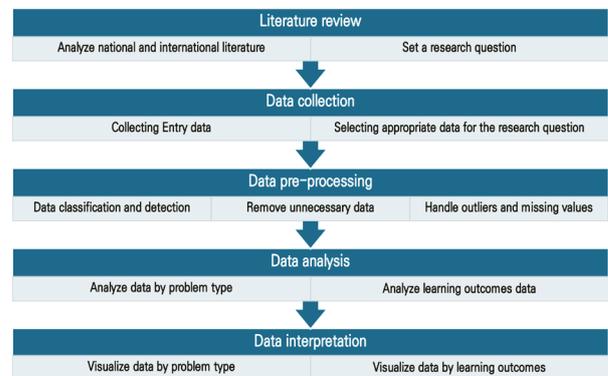


Figure 1. Research Procedure

### 3.1 데이터셋 및 연구 대상

본 연구는 네이버 커넥트재단에서 소프트웨어 (Software, 이하 SW)교육의 활성화와 교육의 균등 제공을 위한 공익목적으로 개발된 엔트리 홈페이지(<https://playentry.org>)에서 수집된 데이터를 제공받아 수행되었다. 엔트리는 누구나 무료로 제공되는 코딩학습 플랫폼으로, 드래그 앤드 드롭(Drag and drop) 방식의 블록 기반의 교육용 프로그래밍 언어이다.

엔트리는 ‘안녕 엔트리봇’, ‘엔트리봇 미션’, ‘인공지능 체험’, ‘발견’의 총 4가지의 교육 콘텐츠를 제공하며, 그 중, ‘발견 콘텐츠’는 엔트리 홈페이지에서 학습자들에게 무료로 개방된 학습 활동으로 그 예시는 Fig. 2와 같다. 학습자는 해당 활동을 통해 활동별로 요구되는 컴퓨팅 요소에 맞추어 가이드가 제공되며, 스스로 문제를 해결할 수 있도록 한다. 이런 활동에 관한 데이터셋을 선정하여 집약적으로 발견 콘텐츠의 주제, 요약, 카테고리, 난이도, 소요 시간 등을 파악할 수 있었다.

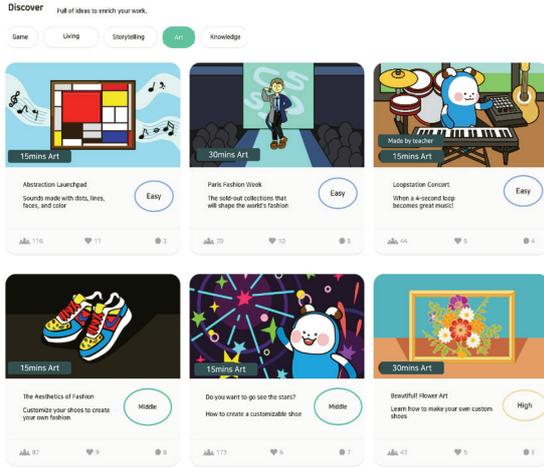


Figure 2. Discover Learning Contents in Entry Website

본 연구를 위한 자료는 2021년 3월 12일부터 2022년 12월 31일까지 수집된 로그 데이터 중 ‘발견 콘텐츠’의 주제, 요약, 카테고리, 난이도, 소요 시간 등과 관련한 데이터 116,927개를 분석에 활용하였다. 수집한 데이터셋의 자세한 내용은 Table 3과 같다.

Table 3. List of Data

Collection Name	Field	Data type	Description
discovery	title	String	the title of learning contents within the entry
	summary	String	summary of learning contents
	categoryCode	String	storytelling, etc, knowledge, arts, game, living
	difficulty	Number	1(low), 2(intermediate), 3(high)
	requiredTime	Number	time spent
	goal	String	learning objectives
	description	String	learning content description
discovery_progress	participantCnt	Number	number of shared collections
	discovery	ObjectID	index for each learning content
	isComplete	Boolean	completion status of learning stages (usually 1 to 10)
	lastIndex	Number	last stage performed by the learner
	status	Array	log data of learning stages
	updated	Date	date last modified by the learner
	created	Date	date of the learner's first content access

### 3.2 자료 분석 방법

#### 3.2.1 EPL 기반 학습에서의 문제 유형화

본 연구는 EPL 기반 학습의 문제를 유형화하기 위하여 Jonassen[11]의 문제 해결의 설계 이론에서 제안된 열한 가지 문제 유형 토대로 학습 콘텐츠를 문제 유형별로 분류하였다. 더불어 타당성 있는 문제 유형 분류를 위해 Lincoln과 Guba[27]와 Cresswell과 Miller[28]가 제시한 질적 연구의 타당성 전략 중 하나인 자료의 다각화와 동료 검토 혹은 보고 및 집중적인 관찰을 수행하였다.

이를 위해 총 30개의 엔트리 학습 콘텐츠의 데이터를 기반으로 Jonassen[11]의 11가지 문제 유형을 분류하였다. 그는 문제 유형을 구분할 때 아래 6가지 기준을 참고하였다. 첫째, 학습 활동은 엔트리의 ‘발견하기’ 학습 콘텐츠 웹페이지에서 주된 활동 과제로 제시한 내용을 참고하였다. 둘째, 성공기준은 각 학습 콘텐츠별로 제시된 학습 목표 내용을 참고하였다. 셋째, 투입물은 프로그래밍 문제 해결이라는 점을 고려하여 각 학습 콘텐츠의 문제 해결에서 요구되는 컴퓨팅 사고(Computational thinking, CT) 요소가 제시된 내용을 참고하였다. 넷째, 맥락과 다섯째, 구조성 그리고 여섯째, 추상성에 관한 부분은 Jonassen이 문제유형 별로 정의한 내용을 참고하여 분류하고자 하였다. 해당 작업에 참여한 총 3인의 연구진은 합의 도출 과정(Consensus-Reaching Approach)을 통해 학습 콘텐츠를 분류하였으며, 의견이 불일치하는 부분에 대해서는 논의를 통해 조정하였다.

이후, 연구진이 분류한 학습 콘텐츠 각각의 유형이 적절하게 분류되었는지를 검토하기 위해 전문가 3인을 대상으로 엔트리 학습 콘텐츠 기반 문제 유형 분류에 관한 타당도를 검토받았다. 첫 번째 전문가는 AI 융합교육 전문가이며, 엔트리, 스크래치 및 블록코딩 교육 관련 교육 진행 및 교안 개발 등 10년 이상의 현장경력을 보유하고 있다. 두 번째 전문가는 전산학전공자로 학교 현장에서 정보 교과 교사로서 근무하고 있으며, 20년 이상의 SW교육 경력을 보유하고 있다. 세 번째 전문가는 컴퓨터교육 및 교육공학 전문가로 약 10년 간 엔트리와 같은 블록 코딩을 포함한 SW교육을 진행하고 관련 연구를 수행한 경력이 있다. 전문가 검토 결과, 전문가 내용 타당도(Content Validity Ratio)를 계산했을 때, 전체적인 전문가 간 일치도(Inter-Rater Agreement)는 0.9로 다소 높은 편이다.

#### 3.2.2 EPL 기반 학습에서 문제 유형에 따른 학습 데이터 분석

본 연구는 문제 유형에 따른 학습자 데이터 분석을 위하여 4.3.3 버전의 R 프로그래밍을 활용하였다. 먼저 데이터 전처리 과정에서 총 37개의 “발견하기” 학습 콘텐츠를 기준으로 데이터를 분류하여 효율적인 분석을 준비하였다. 이후, 불필요한 데이터 항목을 삭제하였다. 각 학습 콘텐츠 데이터셋에는 문제 유형 속성을 파악하기 위한 주제,

요약 설명, 난이도 등의 정보 외에 해당 콘텐츠의 언어, 방 문자 수, 좋아요 개수, 댓글 수, 데이터 식별 번호 등이 포함되어 있었으나, 분석할 수 없는 데이터로 판단된 항목은 분석 범위에서 제외하였다. 특히, 댓글과 같은 정보는 학습 콘텐츠에 대한 학생들의 참여 맥락을 보여줄 수 있는 데이터로 활용 가능했으나, 단순 누적 수치만 제공되었기 때문에 의미 있는 분석이 어려웠다.

최종적으로, 문제 유형화 작업에 필요한 데이터를 선별한 후 이상치가 있는지 확인하고 결측값을 처리하였다. 예를 들어, 학습 콘텐츠에 접속한 횟수를 나타내는 participantCnt 변수에서 값이 '0'인 7개의 학습 콘텐츠를 결측치로 분류하여 제거하였다. 그렇게 30개의 학습 콘텐츠가 남은 상태에서, 각 콘텐츠의 최초 접속일(created 일자)과 마지막 접속일(updated 일자)을 기반으로 참여자들의 학습 콘텐츠 내 활동 지속 시간을 계산하였다. 이후, 전체 시간을 기준으로 사분위수 기반(Interquartile Range) 분석을 수행하여 제1사분위수와 제3사분위수를 크게 초과하는 이상치를 제거하였다.

이후 데이터의 기초 통계량을 분석하여 각 문제 콘텐츠마다 학습자의 이수율, 재확인 비율 등을 계산한 것을 의미한다. 이를 위해 제공받은 엔트리의 로그 데이터들과의 관계를 탐색하였으며, 로그 데이터 중에서 학습 콘텐츠의 이수율, 재확인 비율로 개편화할 수 있는 변수를 조합하였다. 구체적으로 분석한 기준에 관한 설명은 다음과 같다.

첫째, 이수율(completion rate)은 학습자가 프로그래밍 교육을 끝까지 이수한 비율을 의미한다. 선행연구에 따르면, 다른 교과목에 비해 프로그래밍 학습에서 초보 학습자의 중도 탈락률이 절반 이상으로 높게 나타난다[29]. 이에 본 연구는 엔트리에서 제공하는 학습 콘텐츠의 상황에서 이수율이 어떠한지 파악하고자 하였다. 이를 위해 전체 데이터셋에서 학습 완료 여부를 나타내는 isComplete 변수의 개수를 비율로 계산하였다.

둘째, 재확인 비율(review rate)은 학습자가 이전에 학습한 콘텐츠를 다시 확인하기 위해 방문한 빈도를 의미한다. 이는 학습자가 어려웠던 부분을 다시 시도할 기회를 제공한다. 선행연구에 따르면, 학습자가 단순히 정확한 답을 추측하기 위해 문제를 다시 점검하는 행동만으로도 프로그래밍 학습 성과에 긍정적인 영향을 미친다고 보고된 바 있다[30].

본 연구에서는 학습 콘텐츠를 여러 번 방문한 흔적을 비율로 계산하였다. 이를 위해 두 가지 조건을 적용하였다. 첫째, 학습 콘텐츠를 최초로 접속한 created 일자와 마지막으로 접속한 updated 일자의 시간이 다르고 날짜도 다른 경우를 고려하였다. 둘째, 학습자가 문제 콘텐츠 내에서 수행한 학습 단계를 나타내는 로그 데이터(status)와 학습자가 마지막으로 활동한 단계를 나타내는 lastindex 값이 다를 경우, 해당 학습 콘텐츠에 다수 접속하였음을 판단하였다.

셋째, 작품 개수(collection count)는 학습자가 프로

그래밍 학습에서 얼마나 능동적으로 참여하는지를 보여주는 지표이다. 이는 학습자가 단순히 지시에 따라 학습을 수행하는 것에 그치지 않고, 흥미를 느끼며 배운 지식을 활용하려는 모습을 나타낸다[31]. 본 연구에서는 학습자들이 엔트리 학습 콘텐츠 홈페이지에서 공유한 작품 개수의 빈도를 산출하였다. 이후, 문제 유형별 학습 콘텐츠의 평균 작품 개수를 계산하여 학습 콘텐츠 1개당 평균 작품 개수를 도출하였다. 이를 통해 학생들이 가장 능동적으로 참여한 학습 콘텐츠를 식별하고자 하였다.

데이터 해석 과정에는 앞서 분석한 것과 같이 학습 콘텐츠를 문제 유형별로 분석한 내용과 해당 내용을 바탕으로 이수율, 재확인 비율, 작품 개수 등의 학습 결과를 설명하고자 하였다. 이를 위해 문제 유형별로 분석한 내용을 표로 시각화하여 해당 학습 콘텐츠의 학습 카테고리, 학습 콘텐츠 명(주제), 학습 난이도, 예상 문제 해결 시간 등을 고려할 뿐만 아니라 직접 학습 콘텐츠를 선택하여 학습을 시도하는 등 문제 해결 맥락이나 코드 활용 양상을 고려하였을 때 열한 가지 문제 유형 중 어디에 부합하였는지 나타내고자 하였다. 이를 통해 엔트리 내 문제 콘텐츠는 열한 가지의 문제 유형 중 어떤 문제 유형이 어떠한 비율로 차지하고 있는지 설명할 수 있었다. 다음으로, 앞에서 여러 학습 콘텐츠를 문제 유형별로 묶었을 때 이수율, 재확인 비율, 작품 개수를 위주로 살펴보고자 하였다.

## 4. 연구결과

### 4.1 EPL 기반 학습에서 문제 유형 분석 결과

본 연구는 엔트리 내에 '발견하기' 메뉴에 있는 37개의 학습 콘텐츠 중 현재 시점을 기준으로 절대 작동하지 않거나 해당 학습 콘텐츠 활동에 참여한 인원이 0에 가까운 콘텐츠를 제외한 후 총 30개의 학습 콘텐츠를 선별하였다. 이를 토대로 Jonassen[11]이 제시한 문제 유형 내용을 바탕으로 학습 콘텐츠를 분류하였다. 그 결과, 열한 가지 문제 유형 중에서 엔트리 학습콘텐츠와 관계있는 문제 유형은 알고리즘 문제(n=7), 논리 문제(n=7), 설계 문제(n=7), 사례 분석 문제(n=1), 규칙 활용 문제(n=5), 이야기 문제(n=3)로 총 열한 가지 중에서 여섯 가지 유형으로 분류되었다. 또한, 구조화와 비구조화 문제 유형을 고려하였을 때 비구조화 문제 유형에 속하는 설계 문제, 사례 분석 문제, 이야기 문제를 제외하고 나머지 문제 유형(n=19, 63.33%)들은 주로 구조화된 문제에 속하는 것으로 나타났다. 아울러, 엔트리 학습 콘텐츠에는 Jonassen[11]이 비구조화 유형으로서 제시한 의사결정 문제, 트러블 슈팅 문제, 진단 해결 문제, 전략적인 성과 문제, 딜레마 문제는 포함되지 않는 것으로 나타났다.

구체적으로, 논리 문제(logical problem)는 퍼즐, 루미 큐브 게임처럼 특정 규칙에 따라 추론하는 문제를 의미한다[11]. Table 4와 같이 해당 문제 유형을 고려하여 학습

활동, 컴퓨팅 사고 요소를 고려한 투입물 그리고 성공기준을 분석한 내용이 나타나 있다. 그 외에 논리 문제의 문제 맥락, 구조도, 추상도에 관하여, 추상적인 과제 맥락에 발견 가능한 수준의 구조도와 추상도를 보이는 특징이 있다. 예로써 논리적으로 스티커를 배치하고 조작하는 문제와 날짜 및 날씨와 같은 변수 데이터를 제어하여 상태를 보여주는 문제다.

Table 4. Logical Problem Contents (n=7)

Contents	Learning Activity	CT Inputs	Success Criteria Example
Arts A	Make a sound when the mouse pointer touches a shape	Selection, Event, Sound	"Express your feelings and thoughts with dots, lines, and faces."
Arts E	Creating flower art using duplicate entries and variables to create flower art	Replication, Variables, Conditions	"Make the flower move in direction and position."
Game B	Hide objects between backgrounds and signal when clicked	Event, Signal	"Hide pictures by changing the size and angle of objects."
Knowledge B	Can move the background to create the effect of the bus moving forward	Coordinates, Selection, Variables	"Use variables to take a different course of travel."
Living D	Can be created using Papago's AI translation technology	AI Translation, Answer Box, Text Box	"Add an AI translation block."
Living E	Create widgets with weather, date blocks, and lists from your entries	Lists, Variables, Text Boxes, Calculations	"Use the Calculate block to create a date widget."
Living F	Clicking on an object creates a duplicate, which you can then paste wherever you like	Variables, Replicas	"Make stickers drag and drop."

논리 문제에 대한 학습자 데이터 분석 결과는 Table 5에 제시되어 있다. 먼저 이수율 분석 결과, 생활과 도구 카테고리 하에 속하는 문제 콘텐츠들의 이수율이 난이도와 관계없이 평균 41.51%로 가장 높았다. 반면, 게임 카테고리에 속한 문제 콘텐츠는 재확인 비율이 약 4.10%로 다른 카테고리보다 다소 높았으며, 평균 작품 개수 또한 278개로 가장 많았다.

Table 5. Logical Problem Type Results (n=7)

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Arts	Low (n=1)	27.65	2.89	49
	High (n=1)	23.36	2.62	20
Game	Low (n=1)	29.63	4.10	278

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Knowledge	Low (n=1)	41.77	3.27	31
Living	Middle (n=1)	41.88	3.38	113
	High (n=2)	41.14	4.39	239

알고리즘 문제(algorithm problem)는 학교에서 가장 흔히 다루지는 문제로서, 수학에서의 긴 방정식을 풀이하는 것처럼 특정 변수와 공식에 따라 문제를 풀이하는 것을 의미한다[11]. Table 6과 같이 해당 문제 유형을 고려하여 학습 활동, 컴퓨팅 사고 요소를 고려한 투입물 그리고 성공기준을 분석한 내용이 나타나 있다. 그 외에 알고리즘 문제의 문제 맥락, 구조도, 추상도에 관하여, 추상적이고 정형화된 과제 맥락에 절차상 예측 가능한 수준의 구조도와 절차적이며 추상적인 추상도를 보이는 특징이 있다. 예로써 종이접기 순서에 따라 절차적으로 작품을 완성해가는 문제와 오디오를 감지한 후 특정 명령을 입력한 뒤에 절차적으로 동작을 실행하는 문제이다.

Table 6. Algorithm Problem Contents (n=7)

Contents	Learning Activity	CT Inputs	Success Criteria Example
Game A	Add a number to a variable for each click and replace the image after a certain number of clicks	Variables	"Change the variables appearance after a certain number of clicks."
Game C	Can be created utilizing audio detection blocks and read blocks	AI Audio Detection, String	"Move the rocket according to the command you give."
Knowledge C	Enter origami steps into a list and use cues to call them in order	Lists, Variables, Signals	"Click the arrows to go to the previous and next step."
Living A	AI recognizes your face, and objects follow you around	AI, Video Detection, Iteration	"Make a mustache follow me to a specific location on my face."
Living B	Create a timer with variables and a Wait block	Data, Operations, Variables	"Start the timer with a signal and sound an alarm when it ends."
Living C	Create a program that randomly answers questions using Ask and Answer blocks and lists.	Random Number, List, Input	"Randomize the answer to your question."
Storytelling E	Put your life data in a table and show it in a chart	Data Analysis, Variables	"Chart your life data table."

알고리즘 문제에 대한 학습자 데이터 분석 결과는 Table 7에 제시되어 있다. 우선 이수율 분석 결과, 게임 카테고리에 속하며 난이도가 낮은 문제 콘텐츠의 이수율이 54.81%

로, 절반 이상의 참여자가 해당 콘텐츠를 이수하여 가장 높은 결과를 보였다. 반면, 지식 공유 카테고리에 속하는 문제 콘텐츠는 재확인 비율이 18.78%로 다른 콘텐츠에 비해 상대적으로 높았으나, 평균적으로 공유된 작품 개수가 약 10개로 매우 적었다.

Table 7. Algorithm Problem Type Results (n=7)

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Game	Low (n=1)	54.81	5.83	567
	Middle (n=1)	28.53	3.00	117
Knowledge	Middle (n=1)	13.76	18.78	10
Living	Low (n=2)	33.52	3.42	64
	Middle (n=1)	27.77	0.33	75
Storytelling	High (n=1)	30.30	2.67	50

이야기 문제(story problem)는 흔히 교과서 혹은 교사가 제시한 특정 맥락에 맞추어 설정된 알고리즘에 따라 해결하는 문제를 의미한다[11]. Table 8과 같이 해당 문제 유형을 고려하여 학습 활동, 컴퓨팅 사고 요소를 고려한 투입물 그리고 성공기준을 분석한 내용이 나타나 있다. 그 외에 이야기 문제의 문제 맥락, 구조도, 추상도에 관하여, 사전에 정의된 요소와 얇은 맥락의 제약이 있으며 잘 정의되어 있으며 절차상 예측 가능한 구조도를 보이고, 추상도는 제한된 상황에 국한된다는 특징이 있다. 예로써 1막에서 3막까지 주인공의 갈등 상황에서 드러나는 요소를 고려하여 스토리를 조직하는 문제와 속담 이야기 속 장면이 자연스럽게 연결되도록 프로그래밍하는 문제가 있다.

Table 8. Story Problem Contents (n=3)

Contents	Learning Activity	CT Inputs	Success Criteria Example
Stroytelling B	Stitching together multiple scenes to tell a story	Scene, Sequence	"Make up a funny story for the next scene."
Stroytelling C	Create artwork that changes weather effects when a set event occurs.	Events, Shapes	"Press key 1 to show rainy weather."
Storytelling D	Talking blocks and scene transitions allow you to create stories	Sequence, Signal, Scene	"The main character is introduced and confronted with a conflict situation."

이야기 문제에 대한 학습자 데이터 분석 결과는 Table 9에 제시되어 있다. 먼저 이수율을 분석 결과, 이야기 문제의 평균 이수율은 42.90%로, 다른 문제 유형에 비해 가장 높은 수준을 보였다. 특히 난이도가 낮은 콘텐츠의 이수율이 48.84%로 가장 높게 나타났으며, 대체로 재확인 비율

과 평균 작품 공유 개수도 콘텐츠 난이도에 비례하여 증가하는 경향을 보였다.

Table 9. Story Problem Type Results (n=3)

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Storytelling	Low (n=1)	48.84	4.56	70
	Middle (n=2)	36.96	4.15	45

규칙 활용 문제(rule-using problem)는 제한된 조건이 아닌, 다양한 절차와 방법을 고려하여 명백한 해결안을 탐색하는 문제를 의미한다[11]. Table 10과 같이 해당 문제 유형을 고려하여 학습 활동, 컴퓨팅 사고 요소를 고려한 투입물 그리고 성공기준을 분석한 내용이 나타나 있다. 그 외에 규칙 활용 문제의 문제 맥락, 구조도, 추상도에 관하여, 목적성이 있고, 학문적이며 현실 세계의 제약이 따르는 문제 맥락을 갖고 있으며 예측 불가능한 성과의 구조를 가지며 추상도는 요구에 기반하는 특징이 있다. 예로써 미사일 발사에 조절하는 HP값 산출 규칙을 활용하여 해당 동작을 구현하는 문제와 중력에 관한 규칙을 활용하여 지구와 달 사이의 중력 작용 원리를 표현하는 문제가 있다.

Table 10. Rule-using Problem Contents (n=5)

Contents	Learning Activity	CT Inputs	Success Criteria Example
Arts D	Create interactive entries using artificial intelligence techniques	AI Model Training, Shapes	"Change the color of the star object by recognizing the color."
Game D	Make the ball bounce while changing direction and adapt game rules to different situations	Repeat, Condition	"Make the ball change direction when it hits a wall."
Game E	Duplication can be used to reorient balls to stack bricks and break down walls	Duplicate, Iteration, Direction	"When the ball hits, delete the duplicate to make the brick disappear."
Game F	Implementing missiles with cloning and variables to create a shooter game	Variable, Duplicate, Nested Iterations	"Use a variable to decrease the HP value."
Knowledge D	Using variables to represent the gravity of the Moon and Earth	Repeat Variables, Conditions	"Use a variable to represent the gravity of the moon."

규칙 활용 문제에 대한 학습자 데이터 분석 결과는 Table 11에 제시되어 있다. 우선 이수율을 분석 결과, 해당 문제 유형에서는 지식 공유와 예술 카테고리의 학습 콘텐츠가 39.40%와 41.19%로 다른 문제 유형에 비하여 상대적으로 높은 이수율을 보였다. 반면, 게임 카테고리의 학습 콘텐츠는 평균 작품 공유 횟수가 다른 카테고리에 비해 높았으나, 이수율은 낮았다.

Table 11. Rule-using Problem Type Results (n=5)

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Arts	Middle (n=1)	41.19	2.81	94
Game	Middle (n=1)	30.23	0.21	248
	High (n=2)	19.57	5.26	519
Knowledge	High (n=1)	39.40	5.30	17

사례 분석 문제(case analysis problem)는 로스쿨에서 다루는 재판 사례처럼 실체는 아니지만, 현실과 비슷한 맥락을 가진 문제를 의미한다[11]. Table 12와 같이 해당 문제 유형을 고려하여 학습 활동, 컴퓨팅 사고 요소를 고려한 투입물 그리고 성공기준을 분석한 내용이 나타나 있다. 그 외에 사례 분석 문제의 문제 맥락, 구조도, 추상도에 관하여, 실제 세계 기반의 제약적인 문제 맥락을 띠고 있으며 비구조화되어 있고, 사례 속 상황을 담고 있다는 특징이 있다. 예로써 학습자가 환경 보호에 관한 사례를 조사한 뒤 이러한 내용이 반영된 장면을 추가하여 지구의 날을 기념하는 작품을 만드는 문제가 있다.

Table 12. Case Analysis Problem Contents (n=1)

Contents	Learning Activity	CT Inputs	Success Criteria Example
Knowledge E	Embed information in scenes and emphasize important details with different effects	Scenes, Text Boxes, Shapes	“Research the need for environmental protection and add scenes to illustrate it.”

사례 분석 문제에 대한 학습자 데이터 분석 결과는 Table 13에 제시되어 있다. 엔트리 학습 콘텐츠에 포함된 사례 문제 유형은 단 한 가지뿐이었다. 그로 인해 결과 분석에 있어서 다른 문제 유형에 비해 제한적이지만, 다른 유형의 Knowledge 카테고리과 비교하였을 때, 이수율은 큰 차이가 없는 반면에 평균 작품 개수는 상당히 낮은 편에 속하였다.

Table 13. Case-analysis Problem Type Results (n=1)

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Knowledge	Middle (n=1)	32.32	3.14	21

설계 문제(design problem)는 주로 실상 속에서도 모호한 목적과 정해지지 않은 해결 경로로 인해 가장 복잡하고 구조화된 성질을 가진 문제이다[11]. Table 14와 같이 해당 문제 유형을 고려하여 학습 활동, 컴퓨팅 사고 요소를 고려한 투입물 그리고 성공기준을 분석한 내용이 나타나

있다. 그 외에 설계 문제의 문제 맥락, 구조도, 추상도에 관하여, 복잡하고 현실 세계의 자유도가 제한된 입력과 피드백이 오갈 수 있는 문제 맥락을 띠고 있으며 비구조화된 문제 상황이라는 특징이 있다. 예로써 다양한 시의 주제와 표현 방식을 고려하여 학습자가 시를 작성하여 한 행씩 나타내는 문제, 그림판으로 옷을 디자인한 후 신호를 통해 패션쇼를 만들어보는 문제가 있다.

Table 14. Design Problem Contents (n=7)

Contents	Learning Activity	CT Inputs	Success Criteria Example
Arts B	Design clothes in Paint and change clothes with reshape blocks	Shape, Event, Signal	“Design clothes in Paint.”
Arts C	Designing patterns using repetition, coordinates, and stamping	Repeat, Coordinates, Stamp	“Stamp repeatedly to create a pattern.”
Arts F	Using lists and variables to create entries to convey the emotion of my poem	Lists, Variables, Text Boxes	“Write a beautiful poem in a text box.”
Knowledge A	Create your own town by placing buildings on a drawn map	Event, Selection	“Draw a map of my neighborhood on the drawing board.”
Knowledge F	Drawing a fantasy animal in Paint and introducing its features using lists and variables	Lists, Variable	“Draw a fantasy animal on the drawing board and write its characteristics.”
Storytelling A	Use speaking and movement blocks to express different emotions	Sequence, Event	“Express happy emotions.”
Storytelling F	Functions can make the exact same code shorter	Lists, Variables, Signals, Variables, Functions	“Add scenes and choices to create a story with multiple endings.”

Table 15. Design Problem Type Results (n=7)

Learning categories	Difficulty level	Completion (%)	Review (%)	Average Collection Count(n)
Arts	Low (n=1)	16.11	2.65	25
	Middle (n=1)	29.86	2.71	45
	High (n=1)	24.07	4.15	11
Knowledge	High (n=1)	39.89	6.39	418
	Low (n=1)	28.14	2.53	63
Storytelling	Low (n=1)	45.58	4.81	233
	High (n=1)	12.58	4.98	21

설계 문제에 대한 학습자 데이터 분석 결과는 Table 15에 제시되어 있다. 이수율 분석 결과, 스토리텔링 학습 카테고리에서 난이도가 가장 낮은 학습 콘텐츠의 이수율이 45.58%로 가장 높았다. 반면 난이도가 상으로 올라갔을 때는 같은 카테고리의 문제임에도 이수율이 12.58%로 급격히 하락하였다. 반면, 지식 카테고리의 난이도 상인 학습 콘텐츠는 평균 작품 공유 횟수가 총 418개로서 설계 문제 중에서 학생들이 가장 많이 작품을 공유하였다.

## 4.2 EPL 기반 학습에서 학습자 데이터 분석 결과

### 4.2.1 학습 카테고리별 분석 결과

문제 유형에 따라 분류한 뒤 엔트리의 ‘발견하기’의 학습 카테고리를 살펴보았다. 그 결과, 게임(Game), 생활과 도구(Living), 스토리텔링(Storytelling), 예술(Arts), 지식 공유(Knowledge) 카테고리에 따라 문제 유형은 편차 없이 균등하게 분포되어 있었다. 자세한 결과는 Fig. 3과 같다.

학습 카테고리별 분석 결과를 바탕으로 앞서 도출한 학습 데이터를 통합한 결과는 다음과 같다.

첫째, 게임 카테고리의 경우, 엔트리에서 알고리즘 문제( $n = 2$ )와 규칙 활용 문제 유형( $n = 3$ ) 콘텐츠를 상대적으로 많이 포함하고 있었지만, 두 가지 문제 유형 모두 구조화된 문제에 해당하므로 비구조화 문제 유형까지 시사점을 도출하는 것에 제한적일 수 있다. 다만, 이수율을 높이는 관점에서는 알고리즘 문제 유형에서 난이도를 다소 높이는 것보다(28.53%) 난이도를 낮춰서 제공하는 것(54.81%)이 크게 효과적이었다. 이를 통해 게임처럼 학습자의 흥미를 촉진할 수 있는 콘텐츠에서 문제 해결의 난이도를 낮추게 된다면 더욱 높은 참여를 이끌 수 있을 것이다.

둘째, 예술 카테고리의 경우 엔트리에서 설계 문제( $n = 3$ )와 논리 문제( $n = 2$ ) 유형 콘텐츠를 상대적으로 많이 포함하고 있었지만, 이수율을 높이는 관점에서 평균적으로 어느 정도 문제가 구조화된 논리 문제(25.51%) 유형이 2.16% 차이를 보이며 비구조화 정도가 강한 설계 문제(23.35%)보다 다소 효과적이었다. 이때, 특히 구조화 문제에서 문제 해결 난이도 또한 낮을수록 이수율이 4.29% 정도 상승하는 경향을 보였다. 다만, 설계 문제의 경우에는 난이도가 오히려 중간 이상일 때 제일 낮을 때보다 평균적으로 10.86% 정도의 차이를 보이며 이수율이 상승했다. 이에 따라 문제의 구조화 정도를 고려하여 난이도를 단순히 낮추기만 하는 것이 아닌, 비구조화 문제의 경우 난이도를 적정 수준 이상으로 조절하는 것이 문제 해결의 몰입을 높일 수 있을 것이다.

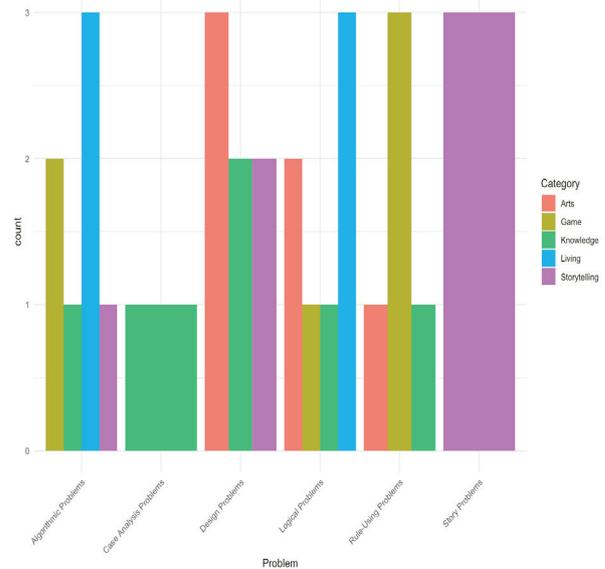


Figure 3. Contents Categories based on Problem Types

셋째, 지식 공유 카테고리의 경우 엔트리에서 이야기 문제 유형을 제외하고 전반적으로 대략 1~2개씩 균형적으로 학습 콘텐츠를 포함하고 있었는데, 그중에서 가장 이수율이 높았던 문제 유형은 설계 문제 유형(39.89%)과 규칙 활용 문제 유형(39.40%)이었다. 규칙 활용 문제는 구조화 문제 범위로 분류하였지만, 그중에서 비구조화 정도가 가장 높은 문제임을 고려했을 때, 비구조화 정도가 높을수록 지식 공유 카테고리에서는 다소 효과적임이 확인되었다.

또한 높은 이수율을 보인 문제 유형 대부분이 그 난이도가 중간 이상이었다는 점에서 지식 공유에서 비구조화된 문제를 제공할 경우 그 난이도를 중간 이상으로 제공할 필요가 있다는 것이 확인되었다. 구조화된 문제 유형의 경우 문제의 난이도가 낮은 경우(34.95%)가 문제 난이도가 중간인 경우(13.76%)보다 높은 이수율을 보여 구조화된 문제의 경우 그 난이도를 낮게 제공할 필요가 있다. 특히 분석한 문제 유형 중에서 가장 비구조화 정도가 높은 설계 문제의 경우, 규칙 활용 문제 유형과 동일하게 난이도가 상으로 높았음에도 불구하고 평균 작품 공유 개수( $n = 418$ )가 다른 문제 유형들의 평균 작품 공유 개수인 24개보다 상대적으로 많았다. 이 점에서 학습자들의 지식 공유 주제의 설계 유형의 문제가 학습자의 능동적인 프로그래밍 문제 해결 능력을 촉진한 것으로 보인다.

넷째, 생활과 도구 카테고리의 경우 주로 알고리즘 문제( $n = 3$ )와 논리 문제( $n = 3$ ) 유형 콘텐츠를 상대적으로 많이 포함하고 있었다. 따라서 두 가지 문제 유형 모두 구조화된 문제에 해당하므로 비구조화 문제 유형까지 시사점을 도출하는 것에 제한적일 수 있다. 다만, 평균적으로 가장 구조화가 된 논리 문제 유형(41.51%)이 알고리즘 문제 유형(30.65%)의 학습 콘텐츠보다 더욱 높은 이수율을 보였다. 이수율뿐만 아니라 학습자들이 평균적으로 작품을 공유한 개수 또한 논리 문제 유형에서 대략 107개 차이로 상

대적으로 높았다. 이때, 난이도의 높낮이에도 큰 편차 없이 지속적인 높은 이수율을 유지했다. 따라서 생활과 도구 카테고리 경우 구조화된 문제 유형의 특성과 조화를 이루며 학생에게 높은 이수율과 능동적 참여를 촉진한 것으로 보인다.

다섯째, 스토리텔링 문제의 경우 대부분 이야기 문제( $n = 3$ ) 유형과 설계 문제( $n = 2$ ) 유형 콘텐츠를 상대적으로 많이 포함하고 있었지만, 난이도에 따른 큰 편차 없이 균형적으로 높은 이수율을 보인 건 평균적으 설계 문제 유형(29.08%)보다 이야기 문제(42.90%)였다. 다만, 이야기 문제와 설계 문제 유형은 동일한 패턴으로 난이도가 낮을수록 각각 이수율이 48.84%와 45.58% 정도로 높아지는 경향을 보였다. 따라서 스토리텔링 문제는 이야기 문제 유형과 조화롭게 연결될 수 있으며, 설계 문제의 경우 난이도를 낮춰서 학습자의 문제 해결 장벽을 낮춘다면 높은 이수율을 촉진할 수 있을 것이다.

#### 4.2.2 학습 난이도별 분석 결과

문제 유형별로 엔트리의 ‘발견하기’ 학습 콘텐츠에서 분류된 학습 난이도(쉬움(low), 중간(middle), 어려움(high)) 분포를 살펴보았다. 자세한 결과는 Fig. 4와 같다.

학습 난이도별 분석 결과를 앞서 도출한 학습 데이터 분석 결과를 통합한 구체적 결과는 다음과 같다.

첫째, 문제 유형에 난이도가 가장 낮은 경우에는 주로 논리 문제( $n = 3$ )와 알고리즘 문제( $n = 3$ ) 그리고 설계 문제( $n = 3$ )가 상대적으로 많이 포함되었는데, 그중에서 이수율이 높은 건 알고리즘의 게임 학습 카테고리의 콘텐츠(54.81%)였다. 그 외에 같은 게임 학습 카테고리의 규칙 활용 문제의 경우에도 대체로 난이도가 낮을수록 이수율이 높아지는 경향을 보였지만, 현재 엔트리에서 규칙을 활용하는 게임 카테고리의 문제에서 낮은 수준의 학습 콘텐츠를 지원하고 있진 않았다. 그 외에 다른 문제 유형에서는 이야기 문제의 스토리텔링 학습 카테고리에 포함된 콘텐츠가 48.84%의 이수율을, 설계 문제의 스토리텔링 카테고리에 포함된 콘텐츠가 45.58%의 꽤 높은 이수율을 보였다. 다만, 해당 경우에 속하는 학습 콘텐츠의 개수는 각각 1개 씩이었다.

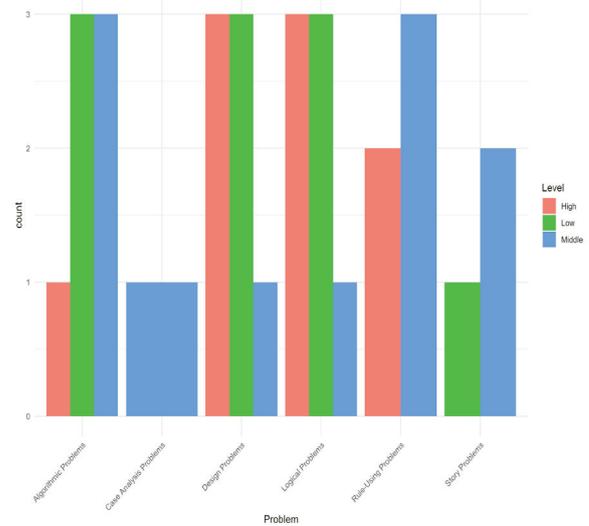


Figure 4. Contents Difficulty based on Problem Types

둘째, 난이도가 중간인 문제 유형은 주로 알고리즘 문제( $n = 3$ )와 규칙 활용 문제( $n = 3$ )가 상대적으로 많이 포함되었는데, 그중에서 이수율이 높은 건 규칙 활용의 예술 학습 카테고리의 콘텐츠(41.19%)였다. 다른 문제 유형에서는 논리 문제 유형의 생활과 도구 학습 카테고리에 속한 학습 콘텐츠가 41.88% 정도의 꽤 높은 이수율을 보였다. 다만, 해당 유형에 속하는 콘텐츠는 하나밖에 없었다.

셋째, 난이도가 가장 높은 문제 유형은 설계 문제( $n = 3$ ), 논리 문제( $n = 3$ )가 상대적으로 많이 포함되었는데, 그중에서 이수율이 높은 경우는 논리 문제의 생활과 도구 학습 카테고리의 콘텐츠(41.14%)였다. 그에 못지않게 다른 유형에서는 설계 문제 유형의 지식 공유 학습 카테고리에 속한 학습 콘텐츠가 39.89% 정도로 꽤 높은 이수율을 보였지만, 해당 경우에 속한 콘텐츠는 하나뿐이었다.

## 5. 결론 및 논의

본 연구는 EPL 기반 학습에서의 학습 문제를 Jonassen[11]의 분류를 기반으로 유형화하고, 학습 데이터를 분석하였다. 이에 따라 엔트리에서 제공하는 학습 콘텐츠를 학습 활동, 투입물, 성공기준, 맥락, 구조성, 추상성을 기반으로 콘텐츠를 분류하였다. 그 결과 Jonassen[11]이 제시한 11개의 문제 유형 중 알고리즘 문제, 논리 문제, 설계 문제, 사례 분석 문제, 규칙 활용 문제, 이야기 문제의 6가지 문제 유형으로 분류하였다. 이후 학습 카테고리와 난이도에 따른 엔트리 학습 데이터를 분류한 문제 유형과 본 연구에서 분석한 의미 있는 지표인 이수율, 평균 공유 작품 개수, 재확인 비율을 고려하여 학습 설계 시사점을 도출하고자 하였으며, 그 내용은 다음과 같다.

학습 주제에 해당하는 카테고리의 관점에서의 학습 설계 시사점을 각 카테고리 별로 도출하였다. 먼저 예술 카테고리의 경우 비구조화된 문제인 설계 문제와 구조화된 문제인 논리 문제 유형을 상대적으로 다수 포함하고 있었다.

각각의 이수율을 분석하였을 때, 낮은 난이도의 구조화 문제와 중간 이상 정도의 비구조화된 문제를 제시하는 것이 학습자들의 문제해결력 향상에 도움이 될 것이다. 또한 게임 카테고리의 경우 구조화된 문제인 알고리즘 문제와 규칙 활용 문제 유형이 상대적으로 많았으며, 이를 통해 낮은 난이도의 구조화된 문제 유형이 학습자들의 문제해결력 향상에 도움이 될 것으로 보인다. 다만, 게임 카테고리에서는 비구조화된 문제에 해당하는 학습 콘텐츠가 없었다는 점에서 비구조화된 문제 유형에서 다양한 난이도를 고려하여 추가적인 학습 과정을 개설하여 그 효과에 대해 분석할 필요가 있다.

지식 공유의 경우 이야기 문제 유형을 제외하고 전반적인 문제 유형을 포함하고 있었으나, 문제의 난이도와 관계없이 비구조화된 문제에서 학습자들의 이수율이 높았다는 점을 기반으로 해당 카테고리에서는 비구조화된 문제 유형을 제공하는 것이 학습자들의 문제해결력 향상에 도움이 될 것으로 보인다. 생활과 도구는 알고리즘 문제와 논리 문제 유형과 같이 구조화된 문제로만 구성되어 있으며, 난이도와 관계없이 높은 이수율을 보임이 확인되었다. 따라서 구조화된 문제에서 현재와 같이 다양한 난이도를 고려하여 학습 과정을 설계하되, 게임 카테고리과 같이 비구조화된 문제에 해당하는 학습 콘텐츠를 다양한 난이도를 기반으로 추가적으로 개설하여 그 효과에 대해 분석할 필요가 있다.

마지막 카테고리인 스토리텔링 카테고리의 경우 비구조화된 문제인 이야기 문제와 설계 문제 유형을 상대적으로 많이 포함하고 있었으며 난이도에 따른 편차가 크지 않았다. 다만 여러 비구조화된 문제 중에서 낮은 난이도의 설계 문제는 특히 높은 효과를 보였다는 점에서 설계 유형에 해당하는 문제를 제공하는 경우 난이도를 고려해서 제시할 필요가 있다.

다음으로, 학습 콘텐츠의 난이도별 데이터를 분석한 결과, 문제 유형의 특성을 고려하여 상, 중, 하의 난이도를 지원할 필요가 있다. 특히 비구조화된 문제 유형일수록 프로그래밍 콘텐츠의 난이도를 적절한 수준으로 조절해야 할 필요가 있다. 문제 해결 상황에서 지나치게 쉬운 문제는 학습자에게 실패보다 지나친 성공의 경험을 강화하여 문제 해결력 향상에 부정적일 수 있다[18-19]. 본 연구에서도 가장 높은 비구조화 정도를 띠는 설계 문제 유형에서 예술 카테고리의 문제 난이도 수준이 제일 낮은 경우 이수율이 16.11% 정도로, 난이도가 중간 수준일 경우(29.86%)와 가장 높은 수준일 경우(24.07%)에 비하여 가장 낮게 드러났다. 따라서 학습자가 비구조화 문제를 탐색하는 과정에서는 의욕을 상실하지 않고, 충분히 의미 있는 학습 경험을 할 수 있도록 적정 수준의 난이도를 제안하는 것이 효과적일 수 있다.

위와 같은 주요 연구 결론을 바탕으로, 전반적으로 EPL 학습 설계에 관한 시사점은 다음과 같다. 첫째, EPL 학습 환경에서는 학습 흥미가 중요한 시작요소이지만[20, 25]., 학습 과정 전반에 있어서는 너무 흥미에 의존하지 않도록

설계할 필요가 있다. 학습자의 흥미에 따라 자신이 경험할 학습 경로가 결정하는 경우 다양한 난이도의 문제나 구조화와 비구조화 문제 경험이 고루 형성되지 못할 가능성이 있다. 그러므로 난이도와 문제의 구조성 정도를 고려하여 학습자에게 부족한 문제 해결 경험을 기반으로 경험하지 않은 콘텐츠를 학습할 수 있도록 제안하는 학습데이터 기반 AI 활용 추천 시스템을 도입하여 다양한 경로를 경험할 수 있도록 도와줄 필요가 있다.

둘째, 엔트리 학습 데이터 분석 결과, 전체적으로 이수율이 낮게 측정되었는데, 이처럼 EPL 기반 학습 환경에서 반복적으로 이수하지 못하는 상황을 경험하는 것은 학습자가 자신의 역량에 대한 부정적인 인식을 초래할 수 있다[32]. 학습자가 자신의 역량을 어떻게 평가하는지에 대한 정의적인 변인인 자기효능감은 컴퓨팅 사고력에 영향을 미치므로[33], 학습자에게 어느 정도 자신의 실력을 긍정적으로 인식할 수 있는 자신에 적합한 수준의 성취 경험을 쌓아가게 하는 것이 필요하다. 따라서 효과적인 EPL 기반 학습을 위하여 학습자의 학습자가 자신의 결과가 아닌 과정을 되돌아볼 수 있도록 질문을 넣거나, 자기 평가와 모니터링을 촉진할 수 있는 도구를 지원하는 등[34], 자기효능감과 자기조절력을 촉진하기 위한 스캐폴딩을 지원할 필요가 있다.

연구결과를 토대로 연구의 제한점과 후속연구를 제안하고자 한다. 첫째, 다양한 EPL 환경에서의 데이터 수집이 필요하다. 본 연구에서 활용된 데이터는 엔트리에서 제공하는 로그 데이터로 엔트리를 활용한 학습자를 대상으로 편의 표집하여 분석하였다. 물론, 엔트리가 국내에서 사용자가 많고, 대표적인 EPL 플랫폼이지만, 그럼에도 본 연구 결과를 모든 학습자 집단에 일반화하여 해석하는 데 무리가 있다. 따라서 본 연구의 일반화를 위하여 다양한 EPL 기반 학습에서 수집되는 데이터를 추가로 분석하여 연구결과를 비교·분석해 볼 필요가 있다.

둘째, 로그데이터를 보완할 다양한 학습 데이터가 필요하다. 본 연구의 데이터는 엔트리에서 관리하는 로그 데이터로 차시별 접속자 ID, 소요시간, 완료 여부 등의 지표들이며, 이를 토대로 학습자의 동기나 학습행동 등을 체계적으로 분석하는 데는 한계가 있었다. 이에 후속연구에서는 더욱 효과적인 학습 데이터 분석을 위하여 학습에 영향을 줄 수 있는 다양한 인지적, 정의적 변인의 데이터 수집이 필요하다.

이와 같은 연구의 제한점에도 불구하고, 본 연구는 EPL 학습에서 학습자가 직면하는 문제 유형을 분석하고, 학습자의 문제해결력을 향상하기 위한 학습 설계 방향성을 분석한 학습 데이터를 기반으로 논의함으로써, 향후 효과적인 EPL 기반 학습 설계를 위한 기초자료를 제공하였다는 점에서 의의가 있다.

## 참고문헌

- [1] Ministry of Education. (2021). *Announcement of the main points of the '2022 revised curriculum'*. Ministry of Education. <https://www.moe.go.kr/boardCnts/viewRenew.do?boardID=294&boardSeq=89671&lev=0&searchType=null&statusYN=W&page=1&s=moe&m=020402&opType=N>
- [2] Son, J., & Kim, T., (2024). Competency Analysis of Elementary Interdisciplinary Informatics Education Focused on Improving Collaborative Problem Solving Ability for the 2022 Revised educational Curriculum. *The Journal of Korean Association of Computer Education*, 27(4), 87-101. <https://doi.org/10.32431/kace.2024.27.4.007>
- [3] Lee, E. A., & Yi, S. H. (2021). The effects of coding education on creative problem solving of academic high school students in creative experience activities. *The Journal of the Korea Contents Association*, 21(6), 716-724. <https://doi.org/10.5392/JKCA.2021.21.06.716>
- [4] Lee, Y. (2018). Analyzing the effect of software education applying problem-solving learning. *Journal of Digital Convergence*, 16(3), 95-100. <https://doi.org/10.14400/JDC.2018.16.3.095>
- [5] Oh, J. H., Jang, D. W., & Chung, I. Y. (2021). The effects of SW education using EPL and coding robot on the computational thinking and problem solving. *Smart Meadia Journal*, 10(3), 60-67. <https://doi.org/10.30693/SMJ.2021.10.3.60>
- [6] Han, Y. S. (2018). Effectiveness of problem-based learning based programming education: Focus on computational thinking. *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, 8(7), 433-445. <https://doi.org/10.35873/ajmahs.2018.8.7.043>
- [7] Kang, E. & Kim, Ji. (2012). Problem-Finding Process and Effect Factor by University Students in an Ill-Structured Problem Situation. *Journal of The Korean Association For Science Education*, 32(4), 570-585.
- [8] Kim, J. & Jo, I. (2024). A Delphi study to develop scaffolding strategies to support complex structural problem solving in web-based PBL environment. *The Journal of Learner-Centered Curriculum and Instruction*, 24(14), 83-107. <https://doi.org/10.22251/jlcci.2024.24.14.83>
- [9] Alshaye, I., Tasir, Z. & Jumaat, N. (2023). The effectiveness of online problem-based learning tasks on riyadh's secondary school students' problem-solving ability and programming skills. *Open Education Studies*, 5(1), 20220208. <https://doi.org/10.1515/edu-2022-0208>
- [10] Lee, C. (2019). *Elementary School Teachers' Difficulties in Learning Programming EPL*, 32(2), 49-63. <https://doi.org/10.24062/kpae.2019.32.2.49>
- [11] Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational technology research and development*, 48(4), 63-85. <https://doi.org/10.1007/BF02300500>
- [12] Hmelo-Silver, C. E., Kapur, M., & Hamstra, M. (2018). *Learning through problem solving*. In International handbook of the learning sciences. Routledge. 210-220. <https://doi.org/10.4324/9781315617572>
- [13] Lee, A. (2018). Domestic research trends analysis of software education. *The Journal of Education Information and Media*, 24(2), 277-301. <https://doi.org/10.15833/KAFEIAM.24.2.277>
- [14] Jun, S. J. (2017). Analysis of research trends and learners' preference for subject area of SW education content. *The Journal of Korean Association of Computer Education*, 20(1), 39-47. <https://doi.org/10.32431/kace.2017.20.1.004>
- [15] Bae, H. J., Lee, E. K., & Lee, Y. J. (2009). A problem based teaching and learning model for scratch programming education. *The Journal of Korean Association of Computer Education*, 12(3), 11-22. <https://doi.org/10.32431/kace.2009.12.3.002>
- [16] Ko, K. (2019). A study on the effectiveness of EPL utilizing programming education based on problem based learning(PBL) for non-SW major. *Convergence Security Journal*, 19(2), 105-111. <https://doi.org/10.33778/kcsa.2019.19.2.105>
- [17] Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational technology research and development*, 67, 541-575. <https://doi.org/10.1007/s11423-018-9622-x>
- [18] Kim, S. (2024). Analysis of learning problems in middle school informatics textbooks: Focusing on problem solving and programming chapter. *The Journal of Korean Association of Computer Education*, 27(1), 113-125. <https://doi.org/10.32431/kace.2024.27.1.009>
- [19] 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. <https://doi.org/10.1007/BF02299613>
- [20] Ahn, H. J., & Ma, D. S. (2013). Development of primary school scratch curriculum for improving the ability to solve problems. *Journal of The Korean Association of Information Education*, 17(3), 317-327. <https://doi.org/10.14352/jkaie.2013.17.3.317>
- [21] Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80, 441-459. <https://doi.org/10.1016/j.chb.2017.09.030>
- [22] Hulls, C. C., & Rennick, C. (2020). Use of a cornerstone project to teach ill-structured software design in first year. *IEEE Transactions on Education*, 63(2), 98-107. <https://doi.org/10.1109/TE.2019.2959591>
- [23] Jiang, B., Zhao, W., Zhang, N., & Qiu, F. (2022). Programming trajectories analytics in block-based programming language learning. *Interactive Learning Environments*, 30(1), 113-126. <https://doi.org/10.1080/104>

94820.2019.1643741

- [24] Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education (TOCE)*, 17(3), 1-25. <https://doi.org/10.1145/3105910>
- [25] Lee, D., & Song, U. Development and Application of Gamification-based English Language Learning Tools Using Entry. *The Journal of Elementary Education*, 39(2), 131-148. <https://doi.org/10.23279/eer.39.2.202406.131>
- [26] Yoon, C., & Kim, T. (2019). Teaching writing in the primary school using Entry software: Focusing on types and aspects of peer feedback. *Primary English Education*, 25(3), 5-30. <https://doi.org/10.25231/pee.2019.25.3.5>
- [27] Lincoln, Y., & Guba, E. G. (1985). *Naturalistic inquiry*. Sage. [http://dx.doi.org/10.1016/0147-1767\(85\)90062-8](http://dx.doi.org/10.1016/0147-1767(85)90062-8)
- [28] Creswell, J.W. and Miller, D.L. (2000) Determining Validity in Qualitative Inquiry. *Theory into Practice*, 39, 124-130. [https://doi.org/10.1207/s15430421tip3903\\_2](https://doi.org/10.1207/s15430421tip3903_2)
- [29] Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM inroads*, 2(4), 71-76. <https://doi.org/10.1145/2038876.2038894>
- [30] Zhang, Q., Zhang, L., Li, B., Chen, L., Hsiao, I., & Wu, F. (2017). Can distributed practice improve students' efficacy in learning their first programming language? *In Proceedings of the 25th International Conference on Computers in Education, ICCE 2017 - Main Conference Proceedings*. Asia-Pacific Society for Computers in Education.
- [31] Kim, J., & Sohn, E. (2021). Difficulty Analysis of an Introductory Computer Programming Course for non-Major Students. *Journal of Creative Information Culture (JCIC)*, 7(2), 69-77.
- [32] Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & education*, 127, 178-189. <https://doi.org/10.32823/jcic.7.2.202105.69>
- [33] Kim, K. (2014). Measuring and applying the self-efficacy in computer programming education. *Journal of The Korean Association of Information Education*, 18(1), 111-120. <https://doi.org/10.14352/jkaie.2014.18.2.255>
- [34] Lishinski, A., & Yadav, A. (2021). Self-evaluation interventions: Impact on self-efficacy and performance in introductory programming. *ACM Transactions on Computing Education (TOCE)*, 21(3), 1-28. <https://doi.org/10.1145/3447378>



## 이지원

- 2022년 강남대학교 교육학과(교육학사), 소프트웨어(공학사)
- 2024년 이화여자대학교 교육공학(석사)
- 2024년~현재 (주)러닝스파크 연구원

✦ 관심분야 : 테크놀로지 기반 학습 설계, 공동설계, 학습과학

✉ jiwon.lee@learningspark.io



## 손예지

- 2023년 이화여자대학교 교육공학과(문학사)
- 2025년 이화여자대학교 교육공학전공(석사)

✦ 관심분야 : 교수설계, 데이터 분석, 이러닝, AI, 문제기반학습

✉ thsakdy0708@ewha.ac.kr



## 이명화

- 2011년 명지대학교 컴퓨터공학과(학사)
- 2015년 이화여자대학교 교육공학과(석사)
- 현재 이화여자대학교 교육공학과 박사과정
- 2022년~현재 강남대학교 기획처 IR센터 책임연구원

✦ 관심분야 : 교수설계, 공학교육, 소프트웨어교육

✉ mlee523@kangnam.ac.kr



## 이정민

- 2001년 이화여자대학교 교육공학과(학사)
- 2003년 이화여자대학교 교육공학과(석사)
- 2009년 플로리다주립대 교육심리 및 교육공학(박사) & 측정 및 통계(석사) 복수학위
- 2009년 퍼듀대학교 연구원
- 2010~현재 이화여자대학교 교육공학과 교수

✦ 관심분야 : 테크놀로지 기반 학습 설계, SW·AI 교육, 학습정서

✉ jeongmin@ewha.ac.kr