

# 컴퓨팅 사고력 측정을 위한 추상화 역량 평가 기준 개발

주여진\* · 마대성\*\*

광주삼정초등학교\* · 광주교육대학교 컴퓨터교육과\*\*

## 요 약

Wing은 21세기를 살아가기 위한 역량으로 컴퓨팅 사고력의 중요성을 역설하였다. 본 연구에서는 컴퓨팅 사고력의 추상화를 관련 연구를 통해 객체 추상화, 데이터 추상화, 절차 추상화로 분류하였다. 또한 추상화 역량 평가를 위해 추상화 성취기준과 평가기준을 개발하였다. 학교 현장에 적용한 결과 학생들은 문제 해결을 위해 글과 그림으로 표현하는 객체 추상화, 데이터 추상화 역량은 우수하지만, 컴퓨터 프로그램으로 작성하는데 어려움을 겪고 있음을 알 수 있었다. 본 연구에서 개발된 추상화 역량 평가는 소프트웨어 교육을 위한 학생 지도에 많은 도움을 줄 것이다.

키워드 : 컴퓨팅 사고력, 추상화, 평가 기준, 소프트웨어 교육, 측정

## The Development of Abstractable Competency Assessment Standards for the Measurement of Computational Thinking

Yejin Ju\* · Daisung Ma\*\*

Gwangju Samjung Elementary School\*

Dept. of Computer Education, Gwangju National University of Education\*\*

## ABSTRACT

Wing, J. M. emphasized the importance of computational thinking competency to live in the 21st century. In this study, the abstraction of computational thinking is classified into object abstraction, data abstraction, and procedural abstraction through related studies. In addition, abstraction achievement and assessment standards were developed for the abstractable competency assessment. As a result of applying to the school site, the students showed that the object abstraction and the data abstraction competency which are expressed in writing and drawing, they are excellent in solving the problem. However, students hard to write a computer program. The abstractable competency assessment developed in this study will help teach students to teach software education.

Keywords : computational thinking, abstraction, assessment standard, software education, measurement

---

이 연구는 주여진(2018)의 석사학위 논문 내용을 보완 연구한 것이다.

교신저자 : 마대성(광주교육대학교)

논문투고 : 2018-06-08

논문심사 : 2018-06-19

심사완료 : 2018-06-27

### 1. 연구의 필요성 및 목적

2015 개정 교육과정에서는 소프트웨어 교육을 강화해야 한다는 국가·사회적 요구를 반영하여 초등학교 실과 교과의 정보통신 활용 교육 내용을 소프트웨어 교육 중심으로 개편하였다. 기존 실과 교과의 정보통신 활용 중심의 정보 단원을 소프트웨어 기초 소양 중심의 대단원으로 구성하여 소프트웨어 도구를 활용함으로써 놀이처럼 재미있게 17시간 이상 학습하도록 하였다[10].

소프트웨어 교육의 목적은 21세기 시민이라면 반드시 가져야 할 컴퓨팅 사고력을 이용하여 문제해결력을 기르는데 있다. 컴퓨팅 사고력은 Wing에 의해 주창되었는데, Wing은 컴퓨터 과학자뿐만 아니라 모두에게 필요한 기본 역량으로 컴퓨팅 사고력을 소개하였다. 컴퓨팅 사고력은 읽기, 쓰기, 셈하기(3R)처럼 아이들이 기본적으로 갖춰야 하는 분석적 역량으로 21세기를 살아가기 위해서는 컴퓨팅 사고력이 반드시 필요하다고 주장하였다[11].

본 연구에서는 2015 개정교육과정의 소프트웨어 교육을 위한 컴퓨팅 사고력 요소들 중 추상화 능력을 평가할 수 있는 방안에 대해 연구하였다. 추상화 역량을 평가하기 위해 추상화 요소를 정의하고, 2015 개정교육과정의 초등 소프트웨어 교육에 적합한 추상화 성취기준 및 평가기준을 개발하고자 한다.

### 2. 관련 연구

#### 2.1 컴퓨팅 사고력

Wing은 컴퓨터 과학자뿐만 아니라 모두에게 필요한 기본 역량으로 읽기, 쓰기, 셈하기(3R)처럼 아이들이 기본적으로 갖춰야 하는 분석적 역량으로 컴퓨팅 사고력을 추가해야 한다고 주장하였다[11].

미국의 CSTA(Computer Science Teacher Association)와 ISTE(International Society for in Education)는 컴퓨팅 사고력에 대한 구성요소를 9가지(자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화)로 세분화

하였다[13].

Google은 컴퓨팅 사고력을 논리적으로 데이터를 정렬 및 분석하고 일련의 과정(알고리즘)을 사용하여 문제를 해결하는 과정과 복잡하고 개방된 문제를 자신있게 다루는 능력이라고 정의하였다[16].

#### 2.2 소프트웨어 교육

2015 개정 교육과정에서는 초등학교 실과 교과의 정보 통신 활용 교육 내용을 소프트웨어 교육 내용 중심으로 개편하였다. 실과 교과의 정보 단원을 소프트웨어 기초 소양 중심의 대단원으로 구성하여 소프트웨어 도구를 활용하여 문제 해결력을 기르도록 하였다. 이에 따른 초등학교 소프트웨어 교육과정 내용 요소는 ‘소프트웨어 이해’, ‘절차적 문제해결’, ‘프로그래밍의 요소와 구조’이며 이에 따른 성취기준은 <Table 1>과 같다[9].

<Table 1> Software Education Achievement Standards

Achievement Standards
[6PS04-07] Find cases where software is applied and understand the impact on our lives.
[6PS04-08] Consider and apply the order of problem solving by procedural thinking.
[6PS04-09] Experience the basic programming process using programming tools.
[6PS04-10] Design a simple program that inputs data, performs the necessary processing, and outputs the results.
[6PS04-11] Understanding the structure of sequence, selection, and repetition in the process of creating a program to solve a problem.

미국의 CSTA(2016)는 초중등단계에서 컴퓨터과학을 가르치기 위한 교육과정 표준안을 발표하였다. 이는 컴퓨터과학 교육과정에 대한 기초를 제공하고, 초중등단계에서 교육할 수 있도록 설계된 핵심 학습목표를 제시하고 있다. 총 네 단계(K-2학년과 3-5학년과 6-8학년과 9-12학년)로 나누었으며 각 단계를 마치는 학생들이 이해하고 수행할 수 있는 하는 개념(Concept)과 활동(Practice)을 표준(Standard)로 표현하였다[13].

영국(Computing at school, 2017)은 국가수준의 컴퓨팅 교육과정(Computing in the National Curriculum)을 제시하고 이는 프로그래밍과 디지털 미디어에서의 창의

적인 작품 제작의 기회뿐만 아니라 컴퓨팅 사고력과 창의성에도 초점을 맞춘다고 명시하고 있다. 영국의 컴퓨팅 교육과정은 컴퓨터 과학(computer science; CS), 정보 기술(information technology; IT), 디지털 리터러시(digital literacy; DL)의 세 가지 측면으로 나누어 구성되었다. 그 중 컴퓨터 과학(CS)는 컴퓨팅 교과학의 핵심으로 학생은 정보와 계산의 원리, 디지털 시스템의 작동 방식 및 프로그래밍을 통해 지식을 사용하는 방법을 학습한다[14].

### 2.3 추상화

Wing(2006)은 컴퓨팅 사고가 추상화와 분해를 통해 복잡한 시스템을 설계하거나 어려운 문제를 해결하는 것이라고 하며 큰 문제를 덩어리로 분리해내는 능력이라고 설명하였다. 복잡한 문제에 접근하기 쉽도록 그것을 적절하게 묘사하거나 관련 있는 특징을 모델링 할 수 있다고 하였다. 더불어 컴퓨팅 사고력의 핵심은 프로그래밍이 아닌 개념화이며 여러 측면에서의 추상화를 통해 사고하는 것이 곧 컴퓨팅 사고라고 주장하였다.

Kramer, J.(2007)는 추상화의 의미를 정확히 하기 위해서 사전적 의미에서 두 가지 측면을 살펴야한다고 하였다. 첫 번째는 ‘어떤 것을 단순화하고 관심 집중을 위해 세부적인 것을 소거하는 과정’이며, 두 번째는 ‘특정 요소의 공통적인 핵심과 본질을 찾기 위해 일반화하는 과정’이다.

CSTA(2016)는 추상화를 과정과 결과물로 나누어 정의하였다. 과정의 측면에서는 ‘주요 아이디어에 집중하기 위해 복잡함을 줄이는 과정. 즉, 문제와 관련 없는 세부 사항을 숨기고, 관련되고 유용한 세부 사항을 모으는 과정을 통해 문제에 집중하는 것’이라고 정의하였다. 결과물의 측면에서는 ‘당면한 문제와 관련 없는 세부 사항을 숨김으로써 문제를 유용하게 재해석하는 사물, 시스템, 또는 문제의 새로운 표현’이라고 정의하고 있다.

정인기(2016)는 추상화는 ‘문제를 해결하기 위해 필요한 자료를 수집·분석하고 적절한 방법(표, 그래프)을 활용하여 파악하기 쉽게 나타내고, 복잡한 요소를 작은 단위로 분해하고, 해결에 필요한 변수들을 추출하여 적절한 해결 모델을 설계하는 것이다.’라고 정의하였다. 학생들이 표현한 추상화 활동 과정을 모델화, 분해, 일반화

로 나누어 상/중/하 세단계 평가기준을 제시하였다[3].

안상진 외(2012)는 문제 해결 측면에서 추상화의 특성을 설명하였다. 첫째, 추상화는 현실 세계에서 일어날 수 있는 문제를 대상으로 한다. 둘째, 실제 문제를 계산 가능한 형태로 추상화한다. 셋째, 문제 해결 방법의 복잡도를 낮추기 위하여 여러 단계를 거쳐 추상화 할 수 있다. 넷째, 추상화된 개념은 정신적 조작을 통하여 문제의 해결책을 수립하는 데 사용한다고 하였다[1].

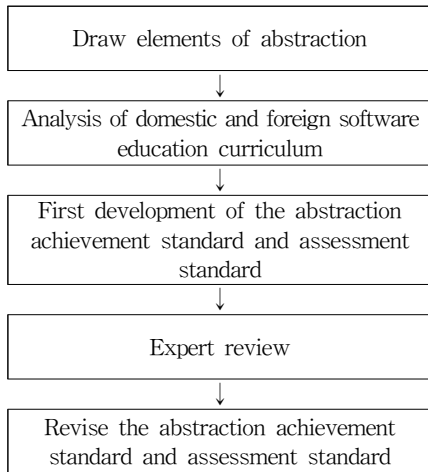
이철현(2016)은 프로그래밍은 눈에 보이지 않는 추상적인 해결 과정을 컴퓨터 화면상의 시각적인 모습으로 나타내는 것이므로 오브젝트의 디자인, 움직임, 소리, 상호 작용, 모양의 변화 등을 표현할 수 있어야 하며 이러한 과정에서 절차와 변수에 대한 추상화가 필요하다고 하였다[7].

위의 연구를 종합해보면 추상화는 복잡한 문제를 해결 가능한 형태로 단순화하는 것을 말한다. 즉, 컴퓨팅 사고력에서 추상화는 컴퓨터가 계산 가능한 형태로 표현되는 것이라고 할 수 있다. 따라서 본 연구에서 추상화는 ‘크고 복잡한 문제를 컴퓨터가 해결할 수 있도록 불필요한 요소를 제거하고 일반화하여 단순한 형태로 만드는 것’이라고 정의한다.

### 3. 추상화 구성 요소

본 연구에서는 선행 연구에 근거하여 추상화의 구성 요소를 도출하고 국내·외 초등 소프트웨어 교육과정을 분석하여 각 추상화 구성요소별 성취기준 및 평가기준을 개발하였다. 전문가 검토를 통해 수정 및 보완 절차를 거쳤으며 추상화 역량 평가기준 개발을 위한 전체적인 절차는 다음 (Fig. 1)과 같다.

컴퓨터 프로그램은 문제를 해결하기 위해 프로그래밍 언어로 추상화된 형태로 존재한다. ‘객체지향’이라는 용어의 의미는 실세계를 해석하여 소프트웨어를 개발하고자 할 때 관점의 주된 대상을 실세계의 객체 위주로 이해하려는 것이다(김태균, 2004; 양승주, 2006). 양승주(2006)는 객체 지향을 이용한 문제해결을 ‘문제 상황’ → ‘객체 분류하기’ → ‘객체 설계’ → ‘객체 간 상호작용’ → ‘문제 해결’의 5단계로 설정하였다. 문제 상황이 주어지면 행위의 주체인 객체를 분류하고, 객체 설계를 통해



(Fig. 1) Development procedure of the abstractable competency assessment standard

분류한 객체의 동작과 속성 값을 결정한다. 객체 분류와 설계가 끝났으면 다른 객체들과의 상호작용을 통해 최종적으로 주어진 문제를 해결한다. 따라서 문제 해결을 위한 프로그래밍의 시작은 문제 상황을 파악하여 필요한 객체를 잘 선별하는 것이다[4][12].

이철현(2016)은 문제 해결 과정을 컴퓨터 프로그램으로 표현할 때 오브젝트의 시각적 표현이나 동작 처리, 변수나 리스트의 이용 등 정교한 수행이 요구된다고 보았다. 프로그래밍은 눈에 보이지 않는 추상적인 해결 과정을 눈에 보이는 컴퓨터 화면상의 시각적인 모습으로 나타내는 것이므로 블록형 프로그래밍 도구를 다룰시 오브젝트의 디자인, 움직임, 소리, 모양의 변화, 상호작용 등을 표현할 수 있어야 한다고 하였다. 또 여러 형태의 자료를 비교하거나 산술적으로 계산하는 연산 수행이 필요하며, 다양한 변인에 따른 수행을 위해 변수나 리스트와 같은 추상적인 개념을 이용해야 한다. 이러한 과정에서 절차와 변수에 대한 추상화가 필요하다고 하였다[7].

위의 연구들을 종합해 보면 추상화의 대상에 따라 객체, 데이터, 절차 측면의 관점으로 구분하여 정의할 수 있다. 본 연구에서는 초등 소프트웨어 교육을 위한 추상화 역량을 객체 추상화, 데이터 추상화, 절차 추상화로 구분하여 각각의 추상화 요소가 초등학교 5~6학년 수준의 소프트웨어 교육에서 어떻게 정의할 수 있는지 알아보고자 한다.

### 3.1 객체 추상화

초등 소프트웨어 교육에서 다루고 있는 교육용 소프트웨어인 스크래치와 엔트리에서는 명령을 실행하는 객체(오브젝트 또는 스프라이트)를 정의하고 그 객체의 행위를 블록을 나열함으로써 나타내고 있다. 따라서 일련의 명령어를 실행하기 위해 객체를 정의하고 속성과 동작을 정의하는 것을 객체 추상화라고 한다[4].

### 3.2 데이터 추상화

문제 해결을 위해 수집된 데이터의 특성을 파악하고 이를 단순화하여 기록하는 과정을 데이터 추상화라고 한다[4]. 초등학생 대상 소프트웨어 교육에서 이러한 데이터를 다루는 데이터 추상화와 관련하여 ‘변수’의 개념으로 설명할 수 있다. 변수는 프로그램에서 사용되는 임의의 값을 일시적으로 메모리에 저장하기 위해 사용된다. 따라서 한 번 선언된 변수와 그 값은 코드 스키프 과정에서 여러 차례 반복되어 사용될 수 있고, 값의 수정이 요구되면 쉽게 일괄적으로 해당 변수값을 바꿀 수도 있다[8].

### 3.3 절차 추상화

절차적 추상화는 문제를 해결하기 위해 필요한 핵심 요소를 추출하고 패턴을 찾아 그에 대한 절차를 나열하는 과정을 말한다. 절차적 추상화를 행하기 위해서 학생들은 컴퓨터가 계산 가능한 형태로 조작하는 과정 즉, 문제 해결의 절차와 패턴을 표현해야 하며 이는 프로그래밍의 순차, 선택, 반복으로 구성된 절차적 과정으로 나타낼 수 있다[4]. 학생들이 문제 해결을 위해 바로 컴퓨터로 표현하는 것은 어렵기 때문에 핵심 요소를 추출하여 절차를 글과 그림(스토리보드, 플로우차트 등)으로 표현해 계획할 수 있다.

## 4. 추상화 성취기준 및 평가기준 개발

추상화 성취기준을 개발하기 위해서 본 논문에서는

초등학교 소프트웨어 교육 수준을 고려하여 연구하였다. 컴퓨팅 사고의 추상화 전체를 다루기에는 그 범위가 매우 크므로 우리나라 2015 교육과정을 토대로 미국의 CSTA 컴퓨터과학 표준 내용, 영국의 국가수준 컴퓨팅 교육과정을 바탕으로 추상화 성취기준을 개발하였다. 개발한 추상화 성취기준은 컴퓨터교육진흥 교수 1인과 석사학위 이상을 소지한 전문가 5인의 검토를 통해 수정 및 보완하였다. 본 연구에서 제시된 10개의 추상화 항목에 대한 검사도구의 알파 신뢰도는  $\alpha = .871$ 로 나타나 신뢰할 수 있음을 알 수 있다.

**4.1. 객체 추상화 성취기준 및 평가기준**

객체 추상화의 성취기준은 문제 상황에서 문제 해결에 필요한 요소만을 객체로 추출하여 표현하는 것이다. EPL이 기반을 둔 객체 지향 프로그래밍은 객체를 중심으로 프로그래밍하며, 객체는 속성과 동작으로 설명할 수 있다. 따라서 학생들이 문제 상황에서 이를 해결하는 프로그램을 만들 때 필요한 객체를 선정하고 이를 속성과 동작을 설정할 수 있어야 한다[2][9][14].

미국의 K-2학년은 속성과 동작에 따라 객체를 분류하는 활동을 하며, 우리나라의 소프트웨어 교육 성취기준 중 [6실04-09]에서는 객체를 속성과 동작으로 표현하여 기초적인 프로그래밍 과정을 체험하도록 하고 있다. 따라서 성취기준을 속성과 동작으로 사물을 설명할 수 있고, 프로그래밍에 필요한 객체를 선정할 수 있으며, 객체에 적합한 속성과 동작을 프로그래밍으로 표현할 수 있는 것으로 객체 추상화의 성취기준과 평가기준을 <Table 2>와 같이 개발하였다.

<Table 2> Achievement and Assessment Standards in Object Abstraction

Achievement Standard	Level	Assessment Standard
[Object-01] Can describe an object using attribute and behavior.	upper	Can correctly understand the attribute and behavior of the object, and accurately describe the attribute and behavior of the selected object.
	middle	Can understand the attribute and behavior of the object, and describe the attribute and behavior of the

Achievement Standard	Level	Assessment Standard
[Object-02] Can select the objects needed to create the program.		selected object.
	lower	Can't understand and explain the attribute and behavior of object.
	upper	Can select all the necessary objects in the process of creating a program.
	middle	Can select roughly the necessary objects in the process of creating a program.
	lower	Can't select the necessary objects in the process of creating a program.
[Object-03] Can use attributes and behaviors appropriate to the object.	upper	Can express attributes and behaviors that are appropriate for an object and describe the interaction between objects.
	middle	Can express attributes and behaviors that are appropriate for an object but can't describe the interaction between objects.
	lower	Can't express attributes and behaviors that are appropriate for an object and describe the interaction between objects.

**4.2 데이터 추상화 성취기준 및 평가기준**

데이터 추상화의 성취기준은 변수에 대한 것이다. 변수는 입력된 자료를 처리하고 출력하는 프로그램을 설계할 때 꼭 필요한 요소이다. 리스트도 데이터를 다룰 때 필요한 개념이지만 세 나라의 교육과정 분석 결과 초등학생을 대상으로는 수준 높은 개념으로 제외했다. 변수의 개념 이해를 토대로 문제 해결을 위한 프로그램에서 필요한 변수를 만들어 프로그래밍에서 변수를 활용할 수 있어야 한다[2][9][14].

<Table 3> Achievement and Assessment Standards in Data Abstraction

Achievement Standard	Level	Assessment Standard
[Data-01] Can understand the notion of variables.	upper	Can explain the notion and necessity of variables.
	middle	Can understand the notion of variables, but can't tell what it needs.
	lower	Can't understand the notion of variables.

Achievement Standard	Level	Assessment Standard
[Data-02] Can select the necessary variables in the program.	upper	Can select all the necessary variables in the process of creating a program.
	middle	Can select roughly the necessary variables in the process of creating a program.
	lower	Can't select the necessary variables in the process of creating a program.
[Data-03] Can use variables in programs.	upper	Can use variables in the process of creating a program.
	middle	Can use variables in the process of creating a program.
	lower	Can't use variables in the process of creating a program.

### 4.3 절차 추상화 성취기준

절차 추상화의 성취기준은 문제 상황을 어떠한 절차로 해결하여 컴퓨터에 추상화된 형태로 입력하는가에 대한 것이다. 학생은 먼저 이야기나 일상생활의 절차를 글과 그림으로 표현하여 일의 순서에 대한 절차를 이해해야 한다. 프로그램을 작성할 때 필요한 알고리즘 순차, 선택, 반복을 사용하여 프로그래밍 할 수 있어야 한다. 더 나아가 자신이 어떻게 문제 해결 프로그램을 만들었는지 설명할 수 있어야 한다[2][9][14].

<Table 4> Achievement and Assessment Standards in Procedure Abstraction

Achievement Standard	Level	Assessment Standard
[procedural-01] In order to solve the problem, only the important elements can be extracted and the problem-solving procedure can be expressed in text and pictures.	upper	The problem solving procedure can be accurately expressed in text and pictures by extracting only the important elements for problem solving.
	middle	The problem solving procedure can be roughly drawn up in text and pictures by extracting only the important elements for problem solving.
	lower	Can't extract only the important elements for problem solving.
[procedural-02] Can list the	upper	In the process of creating a program to solve the problem, the

Achievement Standard	Level	Assessment Standard
statements in order, when they write a program. (sequence)		sequence structure can be presented creatively.
	middle	Can explain the sequence structure in the process of creating a program to solve the problem.
	lower	When they create a program that solves the problem, they can say that there is a sequence structure.
[procedural-03] Can express the problem solving by conditionals. (selection)	upper	In the process of creating a program to solve the problem, the selection structure can be presented creatively.
	middle	Can explain the selection structure in the process of creating a program to solve the problem.
	lower	When they create a program that solves the problem, they can say that there is a selection structure.
[procedural-04] Can find repeated the problem solving steps. (iteration)	upper	In the process of creating a program to solve the problem, the iteration structure can be presented creatively.
	middle	Can explain the iteration structure in the process of creating a program to solve the problem.
	lower	When they create a program that solves the problem, they can say that there is a iteration structure.

## 5. 추상화 역량 평가 및 분석

### 5.1 추상화 역량 평가

추상화 역량 평가기준을 적용하기 위해 교육부 지정 소프트웨어 교육 선도학교인 광주광역시 S초등학교 6학년 30명을 선정하였다. S초등학교는 2015~2016년 소프트웨어 교육 연구학교로 지정되었으며 2017년에는 소프트웨어 교육 선도학교로 지정되어 운영 중이다.

추상화 역량 평가를 위해 컴퓨터교육협의회에서 주관한 2015년도 초등부 경시대회의 출제 문제를 전문가 집단의 자문을 통해 수정하여 검사 도구를 개발하였다. 평가 대상인 S초등학교 6학년 학생들은 이전에 소프트웨어 교육을 받을 때 스크래치로 게임을 만든 경험이 있

었다. 따라서 문제 상황을 ‘스크래치를 이용한 게임 만들기’로 제시하였다.

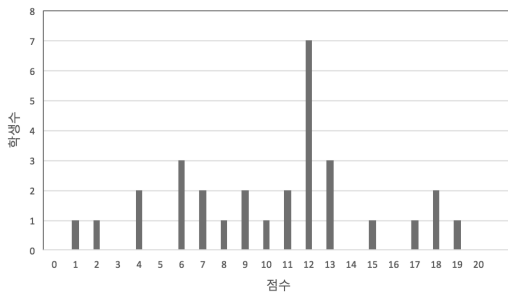
또한 초등학생의 추상화 절차를 평가할 수 있도록 사전 학습지를 문제지와 함께 배부하여 컴퓨터로 프로그래밍하기 전에 글과 그림으로 표현할 수 있도록 하였다.

### 5.2 추상화 역량 평가 분석

본 연구에서 개발한 추상화 평가기준을 바탕으로 학생들이 제시된 문제 상황을 추상화의 과정을 거쳐 프로그램으로 구현할 수 있는지 개인별 평가를 실시하였다. 평가는 상/중/하 점수로 각각 2점/1점/0점으로 변환하고 점수들을 합산하여 각 학생별 추상화 역량이 어느 정도인지 분석하였다.

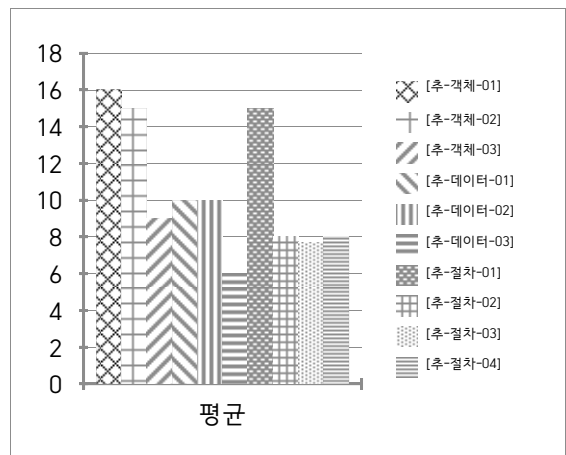
채점자는 소프트웨어 교육 연구학교 및 선도학교 운영 경험이 있는 교사를 선정하였다. 채점자는 평가의 일관성을 위해 1인으로 하였고, 채점에 앞서 본 연구에서 개발한 평가기준을 이해하도록 채점자 훈련을 시행하였다. 채점자는 학생들이 표현하는 과정과 결과물을 관찰 평가로 평가한다. 이 때 본 연구에서 개발한 추상화 역량 평가기준을 체크리스트법으로 학생 개인별 평가를 실시하였다.

학생들의 총점은 1점부터 19점까지 다양하게 분포하였다. 총점을 세 구간(0-6점, 7-13점, 14-20점)으로 구분하여 해당 학생 수를 보면 0-6점은 7명, 7-13점은 18명, 14-20점은 5명으로 나타났다. 평균 점수는 10.4점으로 세 구간 중 중간 점수 구간에 가장 많은 학생이 분포하였다. 추상화 역량 평가에 따른 학생의 점수별 분포는 (Fig. 2)와 같다.



(Fig. 2) Distribution by abstractable competency assessment score (n=30)

추상화 역량을 평가한 결과, ‘중’ 수준에 학생들이 가장 많이 분포함을 확인할 수 있었다. 추상화 성취기준별 평균 점수를 살펴보면 다음과 같다. 첫째, 객체 추상화의 성취기준 [Object-01]은 평균 점수가 16점인 반면 [Object-03]의 평균 점수는 9점에 불과했다. 이는 학생들이 문제 상황에서 객체를 속성과 동작으로 정확하게 설명할 수 있지만 객체를 컴퓨터로 표현하거나, 객체 간 상호작용을 설명하는 것을 어려워함을 알 수 있다. 둘째, 데이터 추상화 성취기준 중 [Data-01]와 [Data-02]의 평균 점수는 10점이지만 [Data-03]의 평균 점수는 6점으로 나타났다. 변수의 개념을 이해하고 필요한 변수가 무엇인지 선정할 수 있지만 프로그램에 변수를 적절하게 활용을 하지 못한 학생이 많다는 것을 알 수 있다. 셋째, 절차 추상화의 성취기준 [Procedural-01]의 평균 점수는 15점으로 대부분의 학생은 문제해결을 위해 중요한 요소를 추출하여 문제해결 절차를 글과 그림으로 표현할 수 있었지만 [Procedural-02]~[Procedural-04]의 평균은 약 8점으로 컴퓨터에 순차, 선택, 반복의 절차로 표현하는 것에 상대적으로 낮은 점수를 보였다.



(Fig. 3) Distribution of each achievement standard average score

따라서 성취기준별 평균 점수 결과를 보면 학생들은 문제 해결을 위해 추상화하는 과정에서 객체와 변수를 선정하고, 절차를 글과 그림으로 잘 표현할 수 있는 반면 이를 컴퓨터로 표현하는 과정에서 어려움을 겪는 경향이 있음을 알 수 있다.

## 6. 결론 및 제언

본 연구에서는 컴퓨팅 사고력에서 추상화를 객체 추상화, 데이터 추상화, 절차 추상화로 분류하고, 이를 평가하기 위해 초등학교 5~6학년 수준의 성취기준과 평가기준을 개발하였다. 추상화 성취기준과 평가기준의 타당도는 전문가 집단의 검토를 통해 개발되었다. 본 연구를 적용하기 위해 광주 S초등학교 6학년 30명을 대상으로 소프트웨어교육을 실시한 후 평가를 실시하였다. 평가 방법으로는 수업에서 과제를 수행하고 있는 학생들을 대상으로 교사가 체크리스트를 이용하여 과정중심 평가를 실시하였다.

성취기준별 평균 점수 결과 분석을 통해 학생들은 문제 해결을 위해 추상화하는 과정에서 객체와 변수를 선정하고, 절차를 글과 그림으로 잘 표현할 수 있는 반면 이를 컴퓨터로 표현하는 과정에서 어려움을 겪는 경향이 있음을 알 수 있었다.

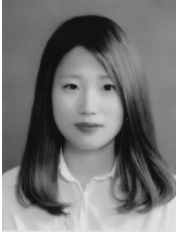
본 연구에서 개발한 성취기준 및 평가기준의 영역별 성취도로 제시된 평가 결과는 소프트웨어 교육을 실시할 때 피드백 자료로서 때 유의미한 자료가 될 수 있을 것이다.

## 참고문헌

- [1] An, S. J, Seo, Y. M, Lee, Y. J (2012). Characteristics of Abstraction in Computational Thinking. *The Korean Association of Computer Education*, 16(1), 43-46.
- [2] CSTA Standards Task Force(2016). K-12 Computer Science Standards Revised 2016, Computer Science Teachers Association.
- [3] Jeong, I. K. (2016). Review of Concept of Abstraction of Computational Thinking. *Journal of The Korean Association of Information Education*, 20(6), 585-596.
- [4] Ju, Y. J, Ma, D. S, Kim, J. R. (2018). A study on abstraction of computational thinking. *Journal of The Korean Association of Information Education*, 9(1), 79-84.
- [5] Kim, H. R, Lee, S. J. (2013). Analysis of computer education courses in foreign countries, KERIS
- [6] Kim, T. G. (2004). Professor K's object-oriented story, Baeumteo.
- [7] Lee, C. H. (2016). Development of Computational Thinking based Problem Solving Model(CT-PS Model) for Software Education. *Journal of Korean Practical Arts Education*, 22(3), 97-117.
- [8] LEE, Y, H, etc. (2011). Processing for fun, Hantee media.
- [9] Ministry of Education (2015). 2015 Revised Practical Arts Curriculum.
- [10] Ministry of Education (2015). Elementary National Curriculum of Korea.
- [11] Wing, J. M. (2006). *Computational Thinking. Communication of the ACM*, 49(3), 33-35
- [12] Yang, S. J. (2006). Object-oriented concept learning in problem solving process using EPL. The Graduate School of Education Korea University, Ed. M.
- [13] CSTA(2016). K-12 Computer Science Framework. Retrieved May 30, 2018, from <https://k12cs.org/>
- [14] Computing at School(2017). Computing in the national curriculum A guide for primary teachers. Retrieved May 30, 2018, from <http://community.computingatschool.org.uk/resources/2618/single>
- [15] Forbes (2017). The world's most valuable brands 2017. Retrieved December 28, 2017, from <https://www.forbes.com/>
- [16] Goggle for Education. Retrieved May 30, 2018, from <https://edu.google.com/resources/programs/exploring-computational-thinking/>



저자소개



**주 여 진**

2018. 광주교육대학교 교육대학원  
졸업(석사)

현재 광주삼정초등학교 교사  
관심분야 : 컴퓨터교육, SW교육,  
프로그래밍

e-mail :

rosemary429@gnue.ac.kr



**마 대 성**

2000 전남대학교 대학원 전산학  
과 졸업(이학박사)

2003~현재 광주교육대학교 컴퓨  
터교육과 교수

관심분야 : 소프트웨어 교육, 정보  
영재교육, EPL 교육, 로봇활  
용교육, 컴퓨팅 사고력

E-mail : dsma@gnue.ac.kr