

Computational Thinking 역량 평가를 위한 서술형 수행평가 도구

전수진 · 한선관

경인교육대학교 컴퓨터교육과

요 약

본 연구는 Computational Thinking 역량을 효과적으로 평가하기 위한 서술형 수행평가 방법에 관한 내용이다. 제안된 서술형 수행평가도구는 MIT에서 제시한 창의컴퓨팅 평가 문항들을 델파이 기법을 통해 조사한 후 문항을 선별 및 수정하였다. 이 서술형 수행평가 도구는 실험하기 및 반복하기, 테스트와 디버깅, 재사용과 재구성, 추상화와 모듈화의 4가지 영역으로 구성되어 총 4문항으로 개발하여 예비교사를 대상으로 수업에 적용하였다. 적용결과 평가 도구는 Cronbach- α 값이 0.6 이상으로 신뢰도가 높았으며 전문가 대상의 델파이 검사 결과 타당도도 높게 나타나 제안된 평가 도구가 학생들의 CT 역량을 효과적으로 측정하는 유용한 평가 도구로 확인되었다.

키워드 : Computational Thinkin, 컴퓨팅 사고, 서술형평가, SW교육 평가, 창의컴퓨팅

Descriptive Assessment Tool for Computational Thinking Competencies

Soojin Jeon · Seonkwan Han

Dept. of Computer Education, Gyeong-in National University of Education

ABSTRACT

In this paper, we proposes a method to effectively evaluate learner competencies of computational thinking. Developed descriptive performance assessment tool is the use of assessment items presented in a creative computing MIT. This tool is selected and modified items through expert delphi technique. The performance assessment tool is composed of four areas that are an experimenting and iterating, testing and debugging, reusing and remixing and abstracting and modularizing. We apply to software classes for the pre-service teachers and conducted this descriptive performance assessment. Applying result of the assessment, the developed assessment tool shows the reliability Cronbach- α values higher than 0.6. In addition, the validity of the test results is higher in experts delphi test. Descriptive assessment tool suggested in this study are identified as a useful assessment tool to effectively measure student CT competencies.

Keywords : Computational Thinking, Descriptive Assessment, SW Education, Creative Computing

교신저자 : 한선관(경인교육대학교 컴퓨터교육과)
논문투고 : 2016-04-18
논문심사 : 2016-04-18
심사완료 : 2016-06-19

1. 서론

교육부와 미래창조과학부는 2014년 SW중심사회실현 정책을 지원하기 위해 「SW중심사회를 위한 인재양성 추진계획」을 수립하였다. 이에 2018년부터는 초중등교육에 SW교육을 실시하며, SW중심 대학을 중심으로 인문계 학생들까지도 SW교육을 실시하겠다고 발표하였다[11].

이러한 시대의 흐름은 SW이 전문 엔지니어를 양성하고자 하는 목적이라고 보다는 보편교육에 목적이 있음을 보여준다[8]. 따라서 SW교육에서 단순 기능의 숙달 보다는 사고력 중심의 Computational thinking(CT)의 역량을 갖추는 것이 중요함을 강조하고 있다[8].

CT의 개념은 J. Wing에 의해 처음 제안되었으며 추상화와 자동화를 핵심 요소로 제시하였으며 이후 다양한 연구자들에 의해 정의되었다[8][13]. 최근 국내에서는 CT를 측정하기 위해 일부 학자들에 의해 평가도구들이 개발되어왔다[5][6][7][12][14]. 이는 CT를 측정할 수 있는 객관식 문항이나 5점 척도식 자기평가 설문문항, 평가를 위한 분석 언어 능력 시스템, 프로그래밍 산출물을 평가하기 위한 평가 루브릭 등이 있다.

그러나 이러한 평가 도구들은 CT를 직접적으로 평가하지 못하거나 학생들의 단편적인 지식이나 결과 중심의 평가에 그치고 있어 보다 더 다양하고 효율적인 SW교육 평가도구의 개발에 대한 더 많은 연구가 필요하다.

본래 CT의 개념은 구글이나 CSTA에서 밝힌바와 같이 컴퓨팅과 관련없는 영역도 상당히 포함하고 있지만 본 연구에서는 Brennan(2012)이 창의 컴퓨팅 교육에서 제시하고 있는 CT 프레임워크에서 제시하고 있는 CT 실습 영역에 초점을 둔 평가도구를 개발하고자 한다.

2. 관련연구

2.1 CT 역량 평가관련 선행연구

최근 CT가 SW교육의 핵심 역량으로 부각됨에 따라 국내에서 일부 CT 평가도구에 대한 연구가 진행되고 있다.

전수진(2015)은 전국 초등학생 1%를 대상으로 실시

한 ICT 리터러시 온라인 평가도구에서 CT를 측정할 수 있는 문항들을 추출하여 검증한 후 결정된 총 43개의 객관식 문항으로 학생들의 CT 수준을 측정하였다. 이 연구에서 CT 문항을 만들기 위한 CT요소로서 Expression of digital information, Logic problems, Problem solving processes, Algorithmic thinking, Branching and iteration, Programming, Modeling 등의 요소를 선정하였다[13].

김수환과 한선관(2012)은 초등학생을 대상으로 정규 모듈의 수업을 구성하여 디자인기반 학습을 통해 스크래치 수업을 실시 후, CT의 컴퓨터적 관점 영역의 인식이 제대로 형성되었는지 알아보기 위해 문항지를 기반으로 하는 자기 설문 평가를 실시하여 검증하였다. 그 자기설문 평가지에는 정보적 문제해결력, self-프로그래밍 능력, self-프로그래밍 흥미도가 포함되어있다[12].

이은경(2009)은 CT 능력 향상 지원을 위해 교육용 로봇을 활용한 로봇 프로그래밍 교수학습 모형을 개발하였다. 개발된 교수학습 모형을 적용하고 학생들의 CT능력 향상정도를 측정하기 위해 OECD/PISA에서 제공하고 있는 평가틀을 활용하였다. 평가의 영역으로 문제발견, 문제분석 및 표현, 문제해결전략의 세 영역으로 제시하여 총 16문항으로 구성하였다[3].

최형신과 김기범(2015)은 스크래치 프로그래밍 수업이 초등예비교사에게 미치는 영향을 CT와 블록의 텍사노미를 활용한 인지적 평가를 통해 측정하였다[6]. 또한, 최형신은 초등예비교사를 대상으로 CT역량을 개발하기 위하여 6개의 CT 세부요소(절차 및 알고리즘, 병행화 및 동기화, 자료표현, 추상화, 문제분해, 그리고 시뮬레이션)를 도출하여 평가 루브릭을 고안하였다[5].

이러한 평가도구들은 학생들의 직접적인 CT능력을 측정하지 못하거나 객관식 문항 또는 산출물 중심의 평가를 하기 때문에 한정된 문항을 이용해야 하거나 결과 중심의 평가에 그치게 될 수 있다는 한계점을 보인다. 또한, 학생들의 관련 지식 및 이해 수준을 넘어선 폭넓은 CT의 과정을 측정하기에는 매우 부족하고 실제 수업 시간에서는 그 많은 문항에 대해 답하거나 기록하기 쉽지 않으며 교사 또한 일일이 채점하고 평가하기 어려워 CT 능력의 측정을 위한 대안평가의 관점으로 CT 기능을 중심으로 서술형 평가 방법 도입이 필요하다.

2.2 창의컴퓨팅 교육 기반 CT 평가도구

MIT의 Brennan(2015)의 스크래치 팀에서는 창의컴퓨팅 가이드북에서 창의 컴퓨팅이라는 의미는 창의성, 역량강화, 컴퓨팅에 관련된 것이라고 제안하였다[9]. 이 가이드북에서 제시하고 있는 CT 프레임워크의 하위 CT요소는 <Table 1>과 같다. 이러한 창의 컴퓨팅 교육에서 제시하는 CT 프레임워크는 블룸이 제시한 교육목표평가 분류에 의해 지식, 기능 태도 평가와 맥을 같이 한다.

본 연구에서는 CT기능에 대한 과정 중심의 평가를 적용하고자 하며, 학생들은 자신 또는 친구들의 창작품, 창의적인 실습에 대해 이야기하는 기회를 제공하는 것을 중요하게 여긴다. 이 안내서에서 제안하는 과정중심의 평가 자료를 수집하기 위한 여러 가지 양식과 다양한 전략을 제시하였다. 이러한 전략은 학생들의 프로젝트에 대한 의견을 음성, 비디오, 텍스트 파일로 기록하거나, 프로젝트 포트폴리오를 평가하고 디자인 저널을 지속적으로 기록하는 것이다.

<Table 1> CT Framework[9]

CT Concept	CT Practice	CT Perspective
Sequence Loops Parallelism Events Conditionals Operators Data	Experimenting and iterating Testing and debugging Reusing and remixing Abstracting and modularizing	Expressing Connecting Questioning

여기서 제공하는 수행평가 도구들은 학생들의 CT의 결과물인 ‘실험하고 반복하기’, ‘테스팅과 디버깅’, ‘재사용과 재구성’, ‘추상화와 모듈화’를 바탕으로 유창성의 발전을 평가할 수 있다고 제안한다.

그러나 기존의 연구들을 살펴보면 CT 개념(Concept) 영역은 프로젝트 산출물 분석[5][12]과 객관식 또는 단답형 필기평가[6] 등의 평가 방법을 사용하여 비교적 객관적인 평가가 가능하지만, CT 실습(Practice) 영역은 수행평가식 시나리오 다자인과 인터뷰와 같은 시간이 오래 걸리고 복잡한 평가 [10]와 자기평가[6] 등이 일부

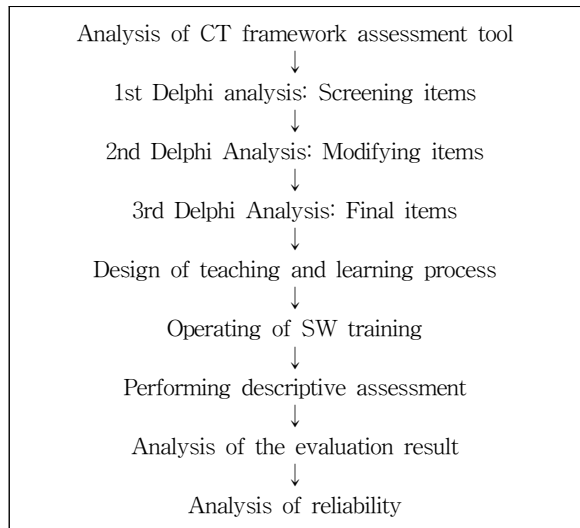
제시되고 있다. 물론, CT 관점(Perspective) 영역은 가장 평가하기 어려운 영역으로써 Brennan(2012)는 자기평가 또는 인터뷰 등의 방법을 제시하고 있다[9][10].

따라서 본 연구에서는 이러한 창의 컴퓨팅 교육에서 제시하고 있는 세 가지 CT 영역 중에서 실습(Practice) 영역에 대한 보다 객관적인 평가 도구를 개발하여 적용하는데 초점을 맞추고자 한다.

3. 연구 방법 및 절차

3.1 연구절차

본 연구에서 제안한 서술형 평가 전략에 대한 연구절차는 (Fig. 1)과 같이 MIT의 창의컴퓨팅에서 제시한 CT 프레임워크를 바탕으로 평가도구 문헌 분석을 먼저 실시하였다. 그리고 3차례의 델파이 분석을 통하여 문항을 선별, 수정한 뒤 확정하였다. 그리고 CT 역량 실기 수업과정을 설계하고 SW교육 강좌에 적용한 뒤 서술형 수행평가를 실시하였다. 평가 결과의 분석을 통하여 평가문항의 신뢰도와 타당도를 확인하는 절차를 거쳤다[1].



(Fig. 1) Research Process

3.2 1차 델파이 분석 : 문항 선별

CT역량 평가를 위하여 먼저 창의 컴퓨팅 가이드북 [9]의 수행평가 목록을 이용하여 SW교육 전문가 10명에게 델파이조사를 실시하였다.

실제로 초보학습자들에게 SW교육을 실시 한 후에 학생들의 CT 역량을 측정하기에 본 서술형 수행평가 문항들이 얼마나 적절한지 5점 척도를 이용하여 선택하도록 하였다. 전문가들은 그들의 교수학습 경험, 평가문항의 명료성, 학생들의 기술 가능성 등을 고려하여 적절성, 활용성 정도를 선택하고 각 문항에 대한 자신들의 의견을 기록하도록 하였다. 이러한 1차 결과의 분석에서는 평균, 사분위값, CVR 값을 분석하여 중요도가 높은 문항을 선별하였다. 그 결과는 <Table 2>와 같다. 그리고 각 문항들에 대한 전문가들의 의견을 바탕으로 불필요한 문항은 제거하거나 평가 항목을 대표하는 문항을 선정하는데 반영하였으며 평가문항의 의도를 명료하게 하도록 사용된 단어나 서술 방식을 수정하였다.

<Table 2> 1st Delphi Analysis: CT Descriptive assesmt items

CT Subject	Item	Avg	CVR
Experimenting and iterating	1. Describe how you built your project step by step.	4.14	0.60
	2. What different things did you try out as you went along with your project?	3.82	0.00
	3. What revisions did you make and why did you make them?	4.27	0.80
	4. Describe different ways you tried to do things in your project, or when you tried to do something new.	4.27	0.80
Testing and debugging	1. Describe what happened when you ran your project that was different from what you wanted.	4.00	0.60
	2. Describe how you read through the scripts to investigate the cause of the problem.	4.00	0.60
	3. Describe how you made changes and tested to see what happened.	4.55	0.80
	4. Describe how you considered other ways to solve a problem.	4.36	0.60
Reusing and remixing	1. Describe if/how you found inspiration by trying other projects and reading their scripts.	4.09	0.40
	2. How did you select a piece of another project, and adapt it for your project?	4.18	0.60

	3. How did you modify an existing project to improve it, or enhance it?	4.55	1.00
	4. How did you give credit to people whose work you built on or are inspired by?	3.18	-0.80
Abstracting and modularizing	1. How did you decide what sprites are needed for your project, and where they should go?	4.27	0.60
	2. How did you decide what scripts are needed for your project, and what they should do?	4.45	0.60
	3. How did you organize the scripts in ways that make sense to you and others?	4.55	1.00

<Table 2>와 같이, 대부분 문항의 전문가 델파이 검사 평균이 3.18~4.64로 나타났으며, 평균 4.21로 대부분 타당한 문항임을 보였다. 또한, 내용타당도비율(CVR)을 통해 정량적인 타당도를 확인하였다. 내용타당도비율(CVR)은 중요도에 대한 합치된 정도를 수치화한 것으로 타당도 검증의 척도로 사용할 수 있다[1]. 패널 수에 따른 CVR의 최소값 기준을 고려할 때, 패널 수가 10명인 경우 CVR값이 .62 이하의 문항은 타당도가 없다고 판단할 수 있다[1]. 따라서, 평균 4.0 이상의 문항과 CVR값이 0.62이상의 문항을 중심으로 문항을 선별 및 수정하였다.

이에 [실험하기 및 반복하기] 영역에서는 문항 3,4의 값이 높기 때문에 이 문항들을 수정하여 만들었다. [테스팅과 디버깅] 영역에서는 3번 문항을 수정하여 만들었다. [재사용과 재구성] 영역에서는 3번 문항을 수정하여 문항으로 만들었다. [추상화와 모듈화] 영역에서는 3번 문항을 이용하여 문항으로 만들었다.

3.3 2차 델파이 분석 : 문항 수정

2차 전문가 델파이 결과의 분석에서는 <Table 3>과 같이 평균, CVR 값을 분석하여 문항의 표현의 적절성에 대한 검증이 필요한 문항에 대한 전문가 패널들의 의견을 반영하는 작업을 하였다. 2차 델파이 검사의 문항 및 각 문항별 채점기준에 해당하는 낮음, 보통, 높음에 대한 전문가 평균값은 3.8~4.7로 높게 나타났다. 그러나 일부 CVR 값이 0.62 이하인 문항 및 채점 기준들은 단어를 수정하거나 의도가 명확하도록 서술방식을 수정하여 표현하였다.

<Table 3> 2nd Delphi Analysis: Modifying items by validation

Category	Delphi	Item	low	mid	high
Experimenting and iterating	Avg	4.60	4.00	3.80	3.90
	CVR	0.8	0.20	0.00	0.40
Testing and debugging	Avg	4.30	4.40	4.30	4.40
	CVR	0.60	1.00	0.80	1.00
Reusing and remixing	Avg	4.50	4.60	4.40	4.40
	CVR	1.00	1.00	0.80	0.80
Abstracting and modularizing	Avg	4.40	4.70	4.20	4.50
	CVR	0.80	1.00	0.60	1.00

3.4 3차 델파이 분석: 서술형 수행평가 문항 확정

위와 같은 결과를 바탕으로 문항을 최종 선택 및 수정하여 CT 항목의 각 영역별로 1문항씩으로 정하고 다음과 <Table 4>와 같이 총 4문항으로 정리하였다.

<Table 4> 3rd Delphi Analysis: Final items

Category	low(1)	mid(2)	high(3)
[Experimenting and iterating] Describe different ways you tried to do things in your project, or when you tried to do something new.	Student provides no examples of trying something new.	Student provides an example of trying something new in the project.	Student describes specific new things(brainstorming, mind mapping) s/he tried in a project.
[Testing and debugging] Describe how you read through the scripts to investigate the cause of the problem and how you made changes and considered the ways to solve a problem.	Student provides problem status simply.	Student provides problem solving way simply.	Student provides problem status by example and solving way specifically.
[Reusing and remixing] Describe how you select a piece of another project(script, idea) and adapt it for your project	Student provides a piece of another project simply.	Student provides an adapting way and piece of another project.	Student provides a piece of another project and adapting way by using specific example.
[Abstracting and modularizing] Describe example and process how you make a script to develop project from real problem.	Student provides about the project development broadly.	Student does not show idea but provides example to develop the project.	Student show idea and provides example to develop the project specifically.

3.5 서술형 수행평가도구의 적용을 위한 교수학습 과정 설계 및 적용

위와 같이 개발된 서술형 수행평가도구를 적용할 수업을 <Table 5>와 같이 설계하였다. 평가는 K-교육대학교 1학년 학생 54명을 대상으로 하였다. 이 학생들에게 한주에 2시간씩 10주차 총 20시간동안 EPL인 스크래치를 이용하여 프로그래밍 수업을 실시하였다. 학생들은 다양한 프로젝트를 분석하고 설계하며 구현하는 경험을 반복하면서 자연스럽게 CT능력을 익혔다.

<Table 5>에서 마지막 항목인 CT개념에서 제공하는 프로그래밍 과학 지식의 기준을 제공하여 평가의 달성 기준을 제시하였다. 채점방식은 기초 1점, 발달 2점, 능숙 3점으로 하였으며 아무것도 기술하지 않은 학생은 0점으로 처리하였다. 모든 항목을 더한 최종 점수를 백분율로 환산하였다.

<Table 5> SW Practice Program for CT Competencies

No	Subject	CT Practice	Program with CT concept
1	Firework	Experimenting and iterating	Sequence, Loops
2	Dance Party	Experimenting and iterating	Sequence, Loops, Parallelism
3	Fishbowl	Experimenting and iterating, Testing and debugging	Loops, Parallelism, Events
4	Wizard Star	Abstracting and modularizing	Mathematic formula in wizard star game. Conditionals, Operators, Data
5	Avoiding Missile	Abstracting and modularizing, Reusing and remixing	Function in an avoiding missile game. Events, Conditionals, Operators, Data
6	Vocabulary	Testing and debugging, Reusing and remixing	Debugging project in vocabulary game. Parallelism, Events, Conditionals, Operators, Data
7	Project Design	Experimenting and iterating,	Individual Free Project and Assessment
8~9	Project Development	Testing and debugging,	
10	Project Share and Assessment	Reusing and remixing, Abstracting and modularizing	Expressing, Connecting, Questioning

또한, 본 연구에서 개선된 서술형 수행평가 도구는 마지막 시간에 온라인으로 문항을 제시하고 40분 동안 모두 함께 4문항에 각각에 모두 기술하여 제출하도록 하였다. 이 평가 도구의 채점 방식도 낮은 수준은 1점, 보통수준은 2점, 높은 수준은 3점으로 채점하였다. 전혀 기술하지 않은 경우에는 0점으로 채점하였다. 이 점수도 모든 문항의 점수를 합산하여 백분율로 환산하였다.

4. 연구결과

4.1 답안 사례

각 문항별로 학생들이 기술한 답안의 사례 중 의미있는 답안을 선정하여 제시하였다.

실험하기와 반복하기: 사용자가 게임의 패턴을 파악하지 못하게 하시 위해 새로운 것들 시도하기 위해 브레인스토밍을 활용했다. 그 예시로 문을 만들어 각각 문제를 내는 문, 다음 층으로 올라가는 문, 미니게임을 실행하는 문을 만들고 층마다 각 문의 위치와 색깔을 달리해서 다시 실행할 때의 지루함을 최소화 시켰고, 문제는 '리스트'를 적용해서 다양한 종류의 문제가 출제될 수 있도록 했다.

테스팅과 디버깅 : 두더지 잡기 게임에서 두더지를 잡아 점수를 얻었는데도 다음 층으로 올라가지 않는 문제점과 반대의 경우인 두더지 점수가 충분하지 못해 생명이 줄어들어야 하는 경우에도 생명이 그대로인 경우를 발견했다. 이것은 이벤트가 발생할 경우 변수 값을 1만큼 바꾸기 또는 -1만큼 바꾸기를 추가하여 수정하였다.

재사용과 재구성 : 다른 프로젝트들과 다르게 프로젝트 내에서 캐릭터가 움직이는 등의 동적인 모습이 부족했다. 이 차이를 다른 팀들이 발표할 때 참고하여 주인공 캐릭터가 문제를 맞혔을 경우 그 문 쪽으로 향하도록 움직이게 했다. 또한 다른 프로젝트에서 최종 보스를 잡는 것을 사격게임으로 했는데 이것을 참고해서 최종 스테이지에 사격게임과 유사한 두더지 잡기 게임을 추가했다.

추상화와 모듈화 : 세공의 경우 보석의 세공 단계 변수를 만들었고, 1단계부터의 초기 세공 확률을 설정해 놓았다. 세공 확률 값이 0에서 특정수까지의 난수보다 큰 경우에 세공이 성공하도록 설정하였다. 예를 들어, 가장 싼 가넷의 1단계 세공의 경우 확률 값을 9로 설정해놓고, '0부터 11까지의 난수' 보다 확률 값인 9가 더 큰 경우에 성공하도록 하여 '0~8'까지의 수가 나오면 세공 성공, 9~11까지의 수가 나오면 세공이 실패하는 방식으로 확률에 따른 성공 실패를 구현하였다. 또한 각 단계마다 확률 변수 값을 줄이며 성공 확률을 낮추는 방식으로 게임의 난이도를 높였다.

이러한 답안은 개발된 평가 문항의 기준에 맞추어 선정하였다.

4.2 평가 결과 분석

우선 평가 도구의 신뢰도를 분석하기 위해 Cronbach의 알파값을 확인해 보니 0.604로 나타났다. 일반적으로 α 값이 0.6이상이면 만족할 만한 신뢰수준이라고 할 수 있기 때문에 본 연구에서 사용된 평가도구의 신뢰도는 적정하였다.

<Table 6> Result of Items test

Item	Female(n=36)		Male(n=27)		Total(n=63)	
	Avg	Std.D	Avg	Std.D	Avg	Std.D
Item 1	2.31	0.71	2.07	0.92	2.21	0.81
Item 2	2.42	0.81	2.07	0.87	2.27	0.85
Item 3	2.28	0.91	1.78	0.97	2.06	0.97
Item 4	2.22	0.76	2.11	0.70	2.17	0.73
Total	9.22	2.09	8.04	2.38	8.71	2.27

문항별 평가 결과는 <Table 6>과 같으며 성별 평가 결과는 <Table 7>과 같다.

<Table 7> Results of gender assessment

Sex	Avg	Std.D	t	D.Fr	Sig
Female(n=36)	76.9	17.4	2.103	61	.040
Male(n=27)	67.0	19.8			
Avg	72.0	18.6			

수업에 적용하여 평가 결과를 분석한 결과, 남녀 차이가 유의미하게 보였다($p < 0.05$). 즉, 서술형 수행평가는 남학생들보다는 여학생들에게 유리한 결과를 가져올 수 있음을 알 수 있다.

이러한 이유는 문장의 표현력이나 결과 제출의 꼼꼼함을 요구하는 서술형 평가에 대한 특징이 여학생들에게 유리하게 나타난 것으로 추론되었다. 수업 현장에서 CT 기능 평가를 할 경우 남녀학생의 특성을 고려하여 평가를 실시해야 한다.

5. 결론 및 제언

본 연구는 초중등 SW교육의 평가방법에 대해 유용한 도구를 제안하였다. 개발된 평가도구는 창의컴퓨팅의 관점에서 개발되었으며 단순한 코딩 스킬이나 프로그래밍 능력보다는 CT의 신장에 대해 평가하는 내용으로 구성되었다. 연구의 결과는 다음과 같다.

첫째, CT 프레임워크의 CT 실습 영역 평가를 위한 수행평가 문항에서 서술형 수행평가 문항을 선정 및 수정하였다.

둘째, 전문가 델파이 검사를 통하여 개발된 평가도구의 타당도가 높게 나왔으며 Cronbach의 알파값을 통해 신뢰도가 적절함을 알 수 있었다.

셋째, 여학생에게 서술형 수행평가가 유리하게 작용할 수 있음을 고려하여 평가도구를 적용하도록 해야 한다. 하지만 반대로 실기 평가에서 충분히 표현하지 못하는 CT 역량을 평가하기 위해 보완적으로 적용하기에 유용할 것이다.

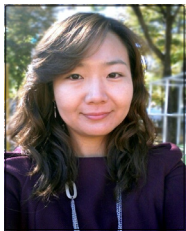
참고문헌

- [1] C. H. Lawshe (1975). A quantitative approach to content validity. *Personnel psychology*, 28(4), 563-575.
- [2] D. K. Jung, M. P. Jung, H. K. Jo, Y. W. No (2014). A Development of the Evaluation Metrics and Analysis of the Object-Oriented Visual Programming Education Using Alice Programming. *Journal of The Korean Association of Information and Communication*, 18(3), 742-748.
- [3] E. K. Lee (2009). Robot Programming Teaching and Learning Model to Enhance Computational Thinking Ability, Korea National University of Education, Doctoral Thesis.
- [4] E. K. Lee (2013). Creative Programming Learning with Scratch for Enhancing Computational Thinking. *Journal of The Korean Association of Computer Education*, 16(1), 1-9.
- [5] H. S. Choi (2014). Developing Lessons and Rubric to Promote Computational Thinking. *Journal of The Korean Association of Information Education*, 18(1), 57-64.
- [6] H. S. Choi, K. B. Kim (2015). The Effects of Scratch Programming on Pre-service Teachers : Assessment Utilizing Computational Thinking and Bloom's Taxonomy. *Journal of The Korean Association of Information Education*, 19(2), 225-232.
- [7] J. B. Song, S. H. Jo, T. W. Lee (2008). The Effect of Learning Scratch Programming on Students' Motivation and Problem Solving Ability. *Journal of The Korean Association of Information Education*, 12(3), 323-332.
- [8] J. M. Wing (2008). Computational thinking and thinking about computing. *Philosophical Transaction of The Royal Society*, 366, 3717-3725.
- [9] K. Brennan, C. Balch, M. Chung (2014). Creative Computing, Harvard Graduate School of Education, Boston.
- [10] K. Brennan, M. Resnick (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.
- [11] Korea Institute of Curriculum and Evaluation (2015). A Study for Developing a Draft of the Informatics Curriculum, Korea Institute of Curriculum and Evaluation Report CRC2015-17
- [12] S. H. Kim, S. K. Han (2012). Design-Based

Learning for Computational Thinking. *Journal of The Korean Association of Information Education*, 16(3), 319-326.

- [13] S. J. Jun (2015). Assessing the ability of computational thinking for elementary school students, Korea University Doctoral Thesis.

저자소개



전 수 진

2000 경인교육대학교(교육학사)
2005 경인교육대학교 교육대학원
(초등컴퓨터교육학 석사)
2014 고려대학교 일반대학원
(컴퓨터교육학 박사)
2000~현재 초등학교 교사
(현 부천부곡초등학교)
관심분야: 초등정보교육, Computational 리터러시, CSCL, SW 교육, 창의컴퓨팅교육, CT
e-mail: soojin.jun@inc.korea.ac.kr



한 선 관

1991 경인교육대학교(교육학사)
1995 인하대학교 교육대학원
(컴퓨터교육학석사)
2001 인하대학교 전자계산공학과
(전산학 박사)
2002~현재 경인교육대학교 컴퓨터교육과 교수
관심분야: 지능형교수시스템, 초등 정보교육, 컴퓨터과학 언플러그드, 게임중독치료, SW교육, 창의컴퓨팅교육, CT
e-mail: han@gin.ac.kr