

Computational Thinking의 개념을 활용한 정보영재 판별도구의 개발

김현수 · 한선관

경인교육대학교 컴퓨터교육과

요 약

본 연구에서 알고리즘 능력을 측정하여 정보 영재를 판별할 수 있는 도구를 개발하였다. 판별 도구는 Computational Thinking의 Concept을 재구성하여 해당 영역이 중점적으로 포함되어있는 문항을 제작하였다. 문항의 유형을 단일 속성 문항, 복합 동일 속성 문항, 복합 집중 속성 문항으로 나누어 세분화하였으며 문제 해결 방식을 전체 스크립트 제작, 스크립트의 일부 수정, 스크립트의 결과 추측하기로 나누어 스크래치 프로그램으로 개발하였다. 개발한 판별도구를 영재학생들에게 적용한 결과 문항의 타당도와 신뢰도가 높은 것으로 나타났다.

키워드 : 교육용프로그래밍 언어, 스크래치, 정보영재 판별도구, 알고리즘 능력, 컴퓨팅 사고

A Development Discrimination Test for Information Gifted Students using the Concepts of Computational Thinking

Hyeonsoo Kim · Seonkwan Han

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

ABSTRACT

This study proposed new approach for the information gifted discrimination test using EPL. We tried to distinguish high level thinking in the information gifted through test of this study. We categorized testing tool according to context type: simple attribute problem, complex same attribute problem, complex focus attribute problem, mathematic-verbal ability required problem. And testing tool divided into modifying scripts, guess scripts's result, using scratch program. In the effectiveness test of proposed tool, we confirmed content validity and credibility.

Keywords : Educational Programming Language, Scratch, Information Gifted Discrimination Test, Algorithm Ability, Computational Thinking

본 연구는 김현수의 2012년도 석사학위 논문(경인교육대학교 교육대학원)에서 발췌하여 정리하였음.

교신저자 : 한선관(경인교육대학교 컴퓨터교육과)

논문투고 : 2015-06-06

논문심사 : 2015-06-09

심사완료 : 2015-08-10

1. 서론

우리나라에서 SW교육에 대한 중요성을 인식하고 초·중등 정식교과로 운영하는 정책과 함께 SW중심사회 실현을 위한 노력을 하고 있다. 또한 수학, 과학 영재와 함께 정보영재를 위한 발굴과 교육에 대한 정책을 추진하여 활발하게 운영되고 있다.

영재 교육을 위해 가장 먼저 필요한 것이 영재를 판별하는 작업이다. 이에 판별 도구의 개발에 대한 연구는 많이 활성화 되었으나 정보영재의 특성 중 하나인 알고리즘 능력을 판별하기 위한 도구는 부족한 편이다.

또한, 알고리즘 능력을 판별하기 위해 사용하는 베이직이나 C언어 등은 초등학생에게는 어려운 프로그래밍 도구이기 때문에 인지부담이 늘어나는 문제점이 있다.

이에 본 연구에서는 정보영재 학생을 판별하기 위해 보다 쉽고 효과적인 교육용 프로그래밍언어를 활용하여 프로그래밍에 익숙하지 않은 학생들이 쉽게 주어진 알고리즘 문제를 창의적으로 해결하도록 하고 이를 측정·분석함으로써 정보 영재를 판별할 수 있는 판별도구를 개발하고자 한다.

2. 이론적 배경

2.1 정보영재의 판별

영재 판별은 영재 교육의 과정으로서 영재를 발견하고 선발하는 기능과 함께 영재를 지속적으로 관리하기 위한 작업이다.

국내의 정보영재학교와 영재교육원의 판별과정을 보면 대부분 다단계 전형을 하고 있다. 국내는 <표 1>과 같이 주로 1단계 서류전형, 2단계 창의적 문제해결 검사, 3단계 면접 및 심층 분석의 단계로 선발하고 있다.

이혜경과 강홍식(2004)은 초등 정보영재는 중, 고등 정보 영재나 수학, 과학 분야의 영재에 비해 컴퓨터 교육을 체계적으로 접해보지 못한 아동들이 속해있을 수 있고, 경험이 있다하여도 대부분 기초적인 컴퓨터 활용 교육과 인터넷, 게임프로그램 등의 사용 수준에 머물러 있는 경우가 많기 때문에 정보 영재성을 판별하기 어려운 점을 지적하며, 영재 교육이 조기에 판별하여 제대

로 된 교육을 하고자 하는 데 있으므로 최대한 정보영재의 특성을 살펴서 정확히 판별하는 것이 중요하다고 제안하고 있다[9].

<Table 1> Type of Discrimination Test for Gifted Talent

Type	Content
Standard Test	IQ, Creative, Achievement, Aptitude Test
Check list, Survey	Teacher, Parent, Colleague, Student
Recommendation	Teacher, Parent, Colleague, Student
Self Report	Portfolio, Observation, Student Record
Expert Decision	IQ, Creative, Achievement, Aptitude Test

2.2 EPL과 스크래치

교육용 프로그래밍 언어(EPL) 중 대표적인 것이 스크래치이다. 블록으로 스크립트를 완성하고 그림, 음악과 같은 멀티미디어 개체를 삽입하여 프로그래밍을 할 수 있는 교육용 프로그래밍 언어이다.

스크래치 프로그램에 대해 조성환(2008)은 “학습자들은 동기유발은 물론이고, 프로그래밍 교육에서 근본적으로 이루고자 하는 목적인 창조적으로 생각하고 문제를 체계적으로 해결하며, 다양한 방식으로 새로운 형태의 문제를 접근할 수 있는 알고리즘 사고방식을 기를 수 있다”고 하였다[4].

한선관과 김수환(2010)은 스크래치 EPL을 활용하여 수학문제해결력을 신장시키기 위한 연구로 정보와 수학을 통합한 교육프로그램을 개발하여 적용하 사고과정의 시각화와 메타인지 측면에서 효과가 있음을 밝혔다[2]. 한선관(2011, 2014)은 정보 영재 학생들을 위한 교육 전략으로 언플러그드와 EPL을 위한 교육 프로그램을 개발하여 적용하였다[1]. 그 외에 스크래치와 관련된 연구가 다양한 방면에서 진행되었다[3][7][8][10].

이상의 선행연구 분석에서 스크래치는 기존의 프로그래밍 언어에 비해 학생들의 인지적 부담을 줄일 수 있고 프로그래밍 능력 및 학습 몰입 수준에 있어 향상을 보이는 것을 알 수 있으며 수학적 사고와 함께 창의적 문제해결력과 논리 사고력을 향상시키는 데 도움을 주는 것으로 나타나 본 연구에서 스크래치를 활용하여 정보영재를 판별할 수 있는 도구를 개발하였다.

2.3 Computational Thinking

Wing(2006)은 CT를 컴퓨터과학자를 위한 것이 아닌 모든 사람을 위한 근본적인 기술이라고 정의했다. 또한, Computational Thinking은 문제해결, 시스템 설계, 인간 행동을 이해하는 능력을 포함한다고 하고 있다[11].

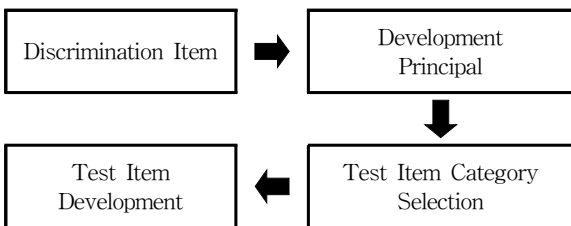
Karen 외(2010)은 CT를 프로그래밍을 통한 CT의 개념과 실재를 이해하기 위해 학습을 기반으로 접근하여 <표 2>와 같이 구조화하였다[5]. 이렇게 구조화된 내용을 바탕으로 학생들의 컴퓨팅 사고를 평가하는 컬러 팔레트 분석 기법을 적용하여 진행하고 있다.

<Table 2> Computational Thinking

Computational Concepts	sequence	loops
	parallelism	events
Computational Practices	conditionals	operators
	variables	lists
	incremental/iterative testing/debugging	
Learning Approaches	abstraction/modularization	
	reuse/remix	
	design	interests
	collaboration	reflection

3. 정보영재 판별 도구의 개발

본 연구에서는 스크래치를 활용하여 정보영재를 효과적으로 판별할 수 있는 판별도구를 개발하였다. EPL을 이용한 정보영재 판별도구의 개발절차는 (그림 1)과 같다.



(Fig. 1) Discrimination Item Development

3.1 판별 요소 설정

본 연구에서는 정보영재의 프로그래밍 능력에 중점을 두고 이를 판별할 수 있는 도구를 개발하고자 한다. 판별 문항을 제작하기 위해 CT의 구성 요소 중 Computational Concepts의 8가지 영역 중에서 Sequence, Loop, Events, Conditions, Operators, Variables, List 영역을 재구성하여 각 영역이 중점적으로 활용될 수 있도록 평가 문항의 기준을 설정하여 제작하였다.

첫째, 판별 도구의 문항은 Karen(2010)이 제시한 Computational Thinking의 Computational Concepts 중 Sequence, Loop, Events, Conditions, Operators, Variables, List 영역을 재구성하여 해당 영역이 중점적으로 드러날 수 있도록 제작한다.

둘째, 학생들이 스크립트를 만들면서 각 영역별로 중점적으로 표현할 블록이 아닌 부수적인 모듈은 문제 안에 제시하고 그것을 활용하도록 문제를 개발한다.

셋째, 본 연구는 정보영재의 프로그래밍 능력을 판별할 수 있는 도구를 개발하는 것이 목적이므로 문항은 학생들이 직접 스크립트를 완성하는 과정을 평가한다.

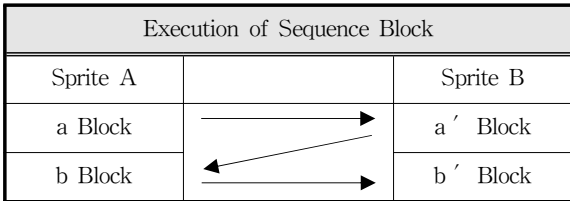
3.2 판별 문항의 구조와 설계

위에서 제시한 정보영재 판별 요소를 바탕으로 학생들의 프로그래밍 능력을 평가하기 위해 문항의 세부 내용을 <표 3>과 같이 설정하였다.

<Table 3> Selection of Test Item

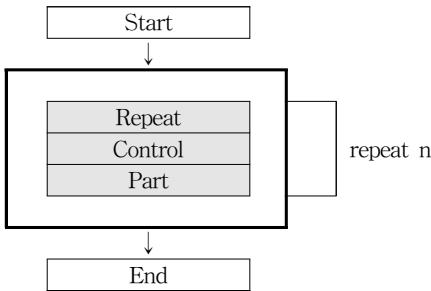
Item	Content
Sequence	Sequence Blocks in Sprite /Thinking Procedure
Loop	Repeat Block for controlling Sprite /Multi shape
Events	Event Control/Watch control
Conditions	Number Comparison Script
Operators	Number Operation in List
Variables	Mouse Control, Pen Script for drawing Web

순차적 구조는 (그림 2)와 같이 스프라이트가 순차적으로 이벤트를 발생시켜 사건이 단계적으로 진행되는 스크립트 구조이다.



(Fig. 2) Structure of Sequence Item

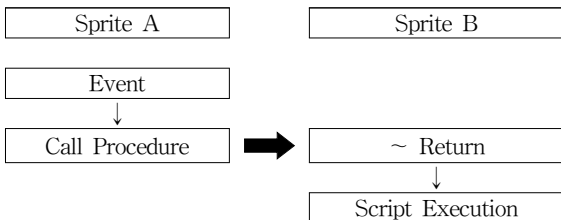
반복은 같은 순차적 구조가 여러 번 반복되어 실행되는 스크립트 구조이다. 반복 구조는 순차적 구조에서 반복되는 부분을 하나로 통합하여 반복 블록을 사용하는 구조이다.



(Fig. 3) Structure of Loop Item

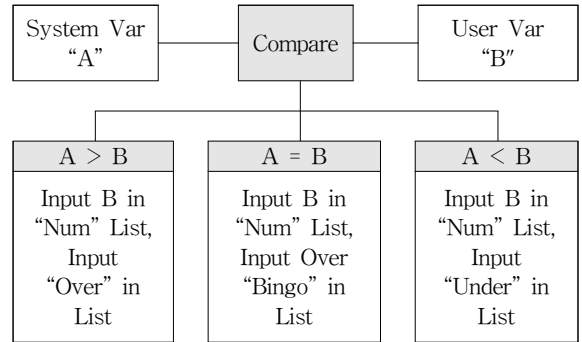
이벤트는 한 스프라이트가 다른 스프라이트의 이벤트나 동작을 야기하는 스크립트 구조이다. 이벤트 구조는 한 스프라이트는 원인이 되는 스프라이트가 이벤트를 발생하기 전까지 대기하고 있다가 정해진 이벤트가 발생되면 동작을 시작하게 된다.

스크래치에서는 이벤트가 발생하는 경우 “방송하기” 블록을 사용하여 이벤트를 발생시키게 되며 “방송하기”를 통해 발생한 이벤트는 “~받을 때”를 이용하여 다른 스프라이트의 동작을 제어한다.



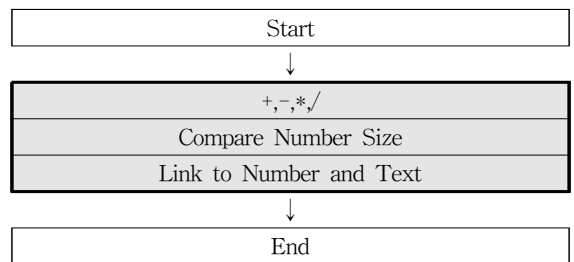
(Fig. 4) Structure of Event Item

조건은 프로그래밍 언어에서 가장 일반적으로 사용되는 조건문이 활용되는 스크립트 구조이다. 변수나 함수 등을 서로 비교하고 판단하여 각각의 조건이 성립될 때 다양한 이벤트가 결정되는 구조이다.



(Fig. 5) Structure of Compare Item

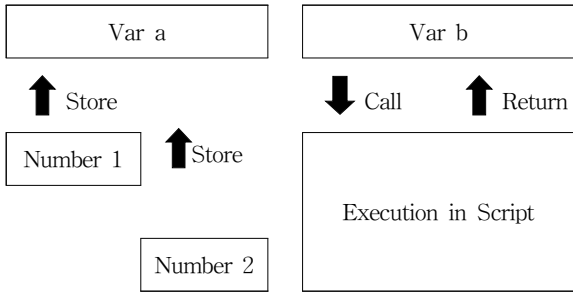
연산자는 스크래치에서 수학적이거나 논리적인 표현을 하기 위해 필요한 연산자 등이 포함된 스크립트 구조이다. 연산자 유형은 독립적으로 사용되기 보다는 다양한 스크립트에서 변수들의 사칙연산이나 크기 비교, 문자의 결합 등을 할 때 부수적인 역할을 하게 된다. 그래서 (그림 6)과 같이 문항의 개발에 있어서도 스크립트의 중간에서 연산자 또는 연결자 역할을 하는 연산자가 중점적으로 부각되어 제시되었다.



(Fig. 6) Structure of Operator Item

변수는 스크래치에서 문자나 항상 변하는 값, 반환 받는 값 등을 저장하고 활용하는 블록이 포함된 스크립트 구조이다.

스크립트 안에서 사용되는 값을 저장하거나 반환 받는 역할이 두드러지도록 문항을 제작하였다.



(Fig. 7) Structure of Variable Item

3.3 판별 문항의 개발

문항을 개발하는데 있어 각 영역에서 사용된 블록 속성의 개수와 다른 속성 블록의 혼합 여부에 따라 3가지로 문항을 구분하였으며 문제의 표현에 있어 수학적 능력을 요구하는 문항과 언어적 능력까지 요구하는 문항으로 구분하였다.

단일 속성 문항은 Concepts의 영역 중 한 가지 블록 속성이 한 번씩만 사용된 문항이다. 이 문항은 본 연구에서 사용된 6가지 속성이 복합적으로 사용되지 않고 최대한 한 가지만 적용될 수 있도록 개발한 문항 유형이다. 그래서 해당 문항은 학생들이 해당 명령어 또는 속성을 이해하고 있는지 판단하는데 활용될 것이다. 본 연구에서는 6문항을 개발하여 적용하였다.

복합 동일 속성 문항은 한 가지 블록 속성이 2번 이상 사용된 스크립트 구조를 가진 문항이다. 이 문항은 본 연구에서 사용된 속성들이 스크립트 내에서 2번 이상씩 활용되므로 이를 통해 학생들의 복합적 사고를 평가하기 위해 2문항을 개발하였다.

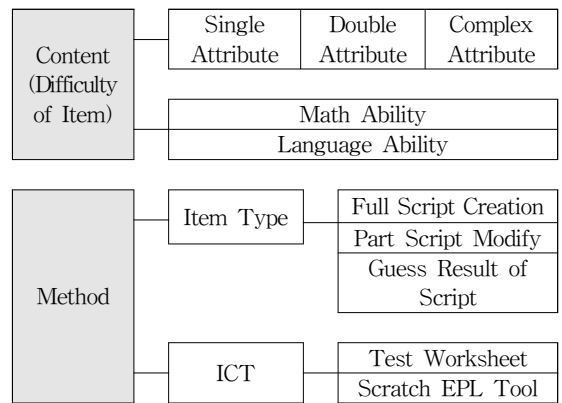
복합 집중 속성 문항은 2가지 이상의 속성 블록이 복합되어 있으며 해당 블록들이 2번 이상씩 사용된 스크립트 구조를 가진 문항이다. 문항을 통해 학생들이 스크립트 전체 맥락을 파악하고 핵심 모듈을 이해할 수 있는지 평가하기 위해 2문항을 개발하였다.

또한, 복합 문항에는 수학적 능력을 요구하는 문항과 수학적 능력을 포함하여 언어적 능력까지 요구하는 문항으로 나눠 개발하였다.

문항을 구성하는 스크립트 내에는 기본적으로 수학적 요소가 반영된 것이 있으므로 해당 문항을 해결하기

위해서는 문항 내에 반영된 수학적 요소를 파악할 수 있는 수학적 사고가 요구될 것으로 보인다. 또한, 문제의 표현을 이해하는 데 있어 수학적 능력을 주로 사용할 수 있는 문항을 개발하였다.

그리고 수학적 능력을 포함하고 언어적 능력까지 요구하는 문항을 제시하여 문항의 난이도를 조절하였다. 문장제 문제를 이해하고 이를 수학적인 알고리즘으로 표현해야 하는 문항을 제시하여 수학적 능력만을 요구하는 문항보다 높은 난이도를 가질 수 있게 하였다.



(Fig. 8) Strategies of Development

본 연구의 판별 문항은 문항을 해결하는 방법적인 측면에 따라 문항의 유형을 3가지로 나누어 개발하였다.

첫째, 학생이 프로그래밍을 직접 하면서 문제를 해결하는 유형이다.

둘째, 주어진 알고리즘과 프로그래밍의 일부분을 제거, 추가하거나 순서를 바꾸게 하는 경우 또는 스크립트를 이루는 블록이나 변수 중 틀린 부분을 발견하게 하고 수정하는 유형이다.

셋째, 제시된 스크립트를 보고 알고리즘의 전체적인 흐름을 파악하고 프로그램의 결과를 추측하는 유형이다.

3.4 판별 문항 개발

CT의 개념을 기반으로 하는 판별 문항을 이용하여 검사지를 개발하기 위해 총 10문항을 개발 추출하였다. 먼저, 연구진에 의해 총 3배수의 30문항을 초기 문항으로 개발하였다. 국내의 컴퓨터 교육 및 컴퓨터 공학 전

문가 9명에게 CT의 개념과 초기 30문항을 제시하고 델파이 조사를 통한 검증 과정을 2회 거쳐 10문항을 선정하고 영재 학생의 선발을 위해 수정, 보완하였다.

본 연구에서 제안하는 정보영재 판별 문항의 실제 개발 예시를 보면 (그림 9)와 같다. 조건을 처리하는 문항지 형식으로 조건처리를 위한 문항이다.

No	4	Domain	Condition, Variable
Difficulty	Complex Focusing Attribute Item		
Ability	Mathematics, Language Ability		
Request	Script Revision Partly		
Type	Worksheet		
Subject	Guessing number that a computer was made		
Problem	Guess a Number Between 1-100 - If too high, then insert "up" in "Up&Down" list - If too low, then insert "down" in "Up&Down" list - And insert that number in "Number" list - If same, then show "Bingo"		
Evaluate	1, 2, 3, 4, 5		

(Fig. 9) Example of Condition Item

4. 연구결과의 적용 및 분석

4.1 대상자 선정

본 연구의 판별 문항의 타당성을 살펴보기 위하여 A 교육청에서 선발되어 영재교육이 진행되기 전의 중학교 정보영재 1학년 학생 25명과 프로그래밍 언어(스크래치)를 배운 적이 없는 일반 학생 25명을 대상으로 개발된 10문항을 적용하였다. 적용된 두 집단은 스크래치를 활용한 프로그래밍 경험이 없는 상태에서 판별 도구를 적용하였다.

<Table 4> Subject of Research

Group	Number
Gifted Talent Group	25
Comparison Group	25

4.2 판별 도구 적용 결과 및 분석

정보영재로 선발된 25명을 대상으로 기존 판별 도구와 본 연구에서 개발한 판별 도구 10문항을 적용하였으며 각각의 결과를 비교하여 유사성을 검증하고자 독립표본 T-검증을 실시하였다. 기존 판별 도구는 김종혜(2008)의 정보과학적 사고 기반의 문제 해결 능력 문항 중 알고리즘적 사고를 검사하는 10문항을 추출하여 활용하였다[3].

본 연구를 통해 검증하고자 하는 가설은 다음과 같이 설정하였다.

[가설 1] 기존의 정보과학적 문제 해결력 능력 판별 결과와 개발한 판별 결과 사이에는 유사성이 있을 것이다.

[가설 2] 일반 학생과 정보영재 학생 간의 결과에는 차이가 있을 것이다.

<Table 5> Result of T-Test (n=25)

Type	Avg	Sdt. Dev	t	p
Exist Test Tool	24.160	5.713	-1.220	.229
Proposed Test Tool	25.920	4.405		

<표 5>와 같이 기존 정보과학적 문제 해결력 판별 도구의 평가 결과를 비교해 보면 t-통계값은 -1.220, 유의 확률(p)은 .229으로 평균 점수 사이에 유의미한 차이가 없는 것으로 밝혀졌다. 통계 분석 결과 기존의 판별 도구와 본 연구에서 개발한 판별 도구의 결과가 유사하여 본 연구에서 개발한 판별 도구의 타당도가 높다고 할 수 있다.

그리고 판별 도구의 신뢰도를 검증하기 위해 내적일관성신뢰도를 측정하였다. 내적일관성신뢰도를 판별하는 지수로 관찰점수 분산과 진점수 분산 비율에 근거한 Cronbach- α 계수를 활용하여 문항내적일관성신뢰도 측정 결과 .882로 나타났다. 일반적으로 Cronbach- α 가 .600 이상이면 신뢰도가 높다고 할 수 있다.

[가설 2]를 검증하기 위해 본 연구에서 개발한 판별 도구를 일반 학생 25명과 정보영재 학생 25명에게 각각 적용하였으며 각 집단별을 판별할 수 있는 도구로 활용할 수 있을지 검증하기 위해 독립표본 T-검정을 실시하였다.

<Table 6> Result of T-test by Group (n=50)

Item	Group	Avg	Std. Dev	t	p
All	Normal	12.720	3.335	11.944	.000**
	Gifted	25.920	4.405		

*p<.05 **p<.001

검증 결과는 <표 6>과 같이 전체 문항 점수 30점에
서 각 집단 간의 평균 점수 차이에 대한 통계적 유의성
을 검정한 결과 t-통계값은 11.944, 유의 확률(p)은 .000
으로 유의 수준 .001에서 유의미한 차이가 있었다. 각
문항별 분석을 보면 1, 2번 문항은 유의미한 차이가 없
었으며 3, 4번 문항은 유의수준 .05 수준에서 유의미한
차이가 있었다. 나머지 문항은 .001의 수준에서 유의미
한 차이를 보여 문항의 복잡성과 난이도에 따라 문항의
판별 특징을 보였다. 종합적으로 분석한 내용을 보면
본 연구에서 개발한 판별 도구는 정보영재의 알고리즘
능력을 판별하는데 활용할 수 있음을 알 수 있다.

5. 결론

정보영재 선발을 위한 판별 도구를 개발하는 연구를
통하여 나타난 결과는 다음과 같다.

첫째, 기존의 정보과학적 사고 기반의 문제 해결 능
력 판별 문항과 본 연구에서 개발한 판별 문항의 결과
를 비교한 결과 유사성 높아 개발한 판별 문항의 타당
도와 신뢰도가 높았다.

둘째, 정보영재 학생과 일반학생에게 판별 문항을 적
용한 결과 두 집단에 유의미한 차이가 있어 제안된 판
별 도구가 영재 판별에 유효함을 알 수 있었다.

본 연구를 토대로 향후 더 많은 대상자에게 적용하여
일반화할 필요가 있지만 CT의 개념을 바탕으로 정교하
게 구성된 콘텐츠를 활용하여 문항을 구성하였기 때문
에 영재 선발에 많은 도움을 줄 것으로 기대한다.

참고문헌

- [1] Han, S. K. (2011). Educational Program for Elementary Information Gifted Student using Unplugged Computing and EPL. *Journal of The Korean Association of Information Education*, 15(1), 31-38.
- [2] Han, S. K. Kim, S. H. (2010). The Comparison of Students Grade Level on the Integrated Learning Program for Mathematical Problem Solving using EPL. *Journal of the Korean Association of Information Education*, 14(3), 311-318.
- [3] Han, S. K. Kim, S. H. (2015). Analysis on the Parents Aware of the Need for the Elementary SW Education. *Journal of the Korean Association of Information Education*, 19(2), 159-168.
- [4] Jo, S. H. Song, J. B. Kim, S. S. Lee, K. H. (2008). The Effect of CPS-based Scratch EPL on Problem Solving Ability and Programming Attitude. *Journal of the Korean Association of Information Education*, 12(1), 77-88.
- [5] Karen, B. Mitchael, R. (2010). Computational Thinking, ScratchEd Webinar Series. <http://scratched.media.mit.edu/resources/computational-thinking-concepts-march-2011-webinar.dia.mit.edu/sites/default/files/BigIdeas.pdf>.
- [6] Kim, J. H. (2008). Secondary Education Program for Problem-solving Ability based on Computational Thinking. Korea University Doctoral Thesis
- [7] Kim, S. H. Han, S. K. (2012). Design-Based Learning for Computational Thinking. *Journal of the Korean Association of Information Education*, 16(30), 319-326.
- [8] Kim, S. H. Han, S. K. (2014). Perceptions of Students who Participate in Scratch Day Events about SW Education. *Journal of the Korean Association of Information Education*, 18(4), 461-470.
- [9] Lee, H. K. Kang, H. S. (2004). Study on the Status of Elementary Gifted and Talented Education

Information. *Injae Journal of Research*, 19(1), 479-502.

- [10] Ryu, M. Y. Han, S. K. (2015). Development of Computational Thinking-based Educational Program for SW Education. *Journal of the Korean Association of Information Education*, 19(1), 11-20.
- [11] Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.

저자소개



김 현 수

2004 경인교육대학교(교육학학사)

2002 경인교육대학교 컴퓨터교육
과(교육학석사)

2015 현재 조산초등학교 교사

관심분야: 초등 컴퓨터교육, 스마
트교육, SW교육, 교육용 프
로그래밍 언어

e-mail: micro82@empal.com



한 선 관

1991 경인교육대학교(교육학학사)

1995 인하대학교 교육대학원(컴
퓨터교육학석사)

2001 인하대학교 전자계산공학과
(전산학 박사)

2002~현재 경인교육대학교 컴퓨
터교육과 교수

관심분야: 창의컴퓨팅교육, SW교
육, 지능형시스템, 초등정보
교육, 디자인기반교육

e-mail: han@gin.ac.kr