

컴퓨터 프로그래밍과 창의성 발현 활동에 관한 질적 사례 연구: NetLogo 기반의 계산적 사고 중심으로

전영국[†]

요 약

본 논문의 목적은 정보과학영재반 학생들이 계산적 사고의 관점에서 NetLogo 프로그래밍을 하면서 문제를 해결하는 과정의 특성을 살펴보는 데 있다. 이를 위하여 전남 지역의 S대 부속 과학영재교육원에서 실시한 중등정보과학 수업에 관한 강의 자료, 학생들의 프로그래밍 산출물, 소감문을 수집하였고 면담 및 학생 활동을 촬영한 비디오 자료를 추가하였다. 연구자는 NetLogo 프로그래밍 수업 중 다각별 그리기 과제에서 나타나는 학생들의 사고 및 프로그래밍 작업 패턴, 학습태도, 행동특성 등을 분석함으로써 계산적 사고와 관련된 프로그래밍 학습의 특징을 도출하였다. 연구참여자(진우)는 계산적 사고를 해 나가는 과정과 컴퓨터 프로그래밍으로 코딩하는 방식에서 번득이는 아이디어의 순간 포착 및 간결한 코딩 등의 양상을 보여주었다. 사례간의 비교고찰과 해석적 논의를 통해 중등정보과학 영재수업을 담당하는 교육실천가들에게 계산적 사고를 함양시키는 프로그래밍 교수-학습과정에 관한 후속 연구 방향을 제시하였다.

주제어 : 계산적 사고, NetLogo 프로그래밍, 질적 사례 연구

A qualitative case study of computer programming and unfolding creative processes: focusing on NetLogo-based computational thinking

Young-cook Jun[†]

ABSTRACT

The aim of this paper is to explore and understand the gifted student's characteristics such as NetLogo programming patterns, attitudes, his/her interest in problems solving. Based on transcripts and coding video frames, we explored the meaningful scenes to come up with thinking patterns, NetLogo programming patterns, attitudes, behaviors on tasks such as drawing regular starlike shapes. This case study contrasts with two other students revealing their unique characteristics both in computational thinking patterns and coding activities. The participant reveals his own ways of finding a clue and elaborating it further for coming up with concise NetLogo coding. This paper provides cross-case discussion and future research direction on how to improve gifted education in terms of problem solving in creative ways.

Keywords : Computational thinking, NetLogo programming, Qualitative case study

[†] 종신회원: 순천대학교 컴퓨터교육과 교수

논문접수: 2014년 12월 21일, 심사완료: 2015년 3월 20일, 게재확정: 2015년 3월 23일

* 본 논문은 2013년 순천대학교 학술연구비 공모과제로 연구되었음

1. 서론

최근 IT 기술이 다양한 과학 분야에서 융합의 추세를 보이면서 컴퓨터 프로그래밍을 통하여 창의적 활동을 지원하는 소프트웨어 활용에 대한 관심이 늘어나고 있다. 각 분야에서 등장하는 문제에 대하여 창의적인 아이디어를 찾아내고 이를 분석적인 사고를 통하여 발전시킨 후에 절차적 사고로 전환하여 컴퓨터 프로그래밍으로 코딩함으로써 문제를 해결하는 능력이 주목받고 있다. LOGO 프로그래밍은 아동들의 문제해결과 관련된 사고력과 창의력을 향상시키는데 도움이 되는 것으로 알려져 있다[9][12][15]. 문제해결 과정은 종종 분석적 사고, 논리적 추론, 알고리즘적 추론, 창의적 사고, 인지적 스킬 등을 요구하는데 이러한 사고 방식을 통틀어 계산적 사고라고 칭하며 컴퓨터 프로그래밍과 더불어 21세기 정보교육과정에서 중점적으로 다루어져야 한다[1][4][5].

최근 연구들은 계산적 사고와 컴퓨터 프로그래밍 활동을 미시적으로 살펴보면서 개인 학생의 특성을 분석하기 위하여 질적 연구 방법을 도입하는 추세를 보여주고 있다. 비디오 촬영과 면담을 사용한 질적 방법을 도입하면 연구자들은 계산적 사고 과정에 대한 면밀한 분석을 통하여 교수-학습 방법을 깊이 있게 탐구할 수 있다. 전영국과 동료들은 하노이탑 문제를 C언어로 해결하는 고등학생의 활동을 비디오로 촬영하고 면담 자료를 분석함으로써 아이디어의 포착과 전개, 알고리즘 디자인과 C 코딩에 관한 미시적인 부분을 해석하였다 [4]. 이러한 연구 방법은 영재교육 분야에 확장되고 있다[10][11].

본 연구의 목적은 정보과학반 영재들의 계산적 사고의 관점에서 NetLogo 프로그래밍을 하면서 문제를 해결하는 과정의 특성을 살펴보는 데 있다. 이를 위해 연구자는 정보와 수학을 융합한 문제해결 과정에서 계산적 사고를 토대로 창의성이 발현되는 과정과 NetLogo 프로그래밍 활동에서 나타나는 특성을 비디오 촬영과 비디오 회상 면담을 통해 질적 사례 연구로 살펴보고자 한다.

2. 이론적 배경

2.1 정보과학 영재교육

영재교육에 관한 연구는 영재들의 정의적 특성이나 행동특성에 관한 연구 외에도 문제해결 과정을 다룬 연구가 보고되고 있다. 영재아와 일반 아동의 문제해결 과정을 비교 연구한 이혜주의 연구는 하노이탑 과제를 수행하는 과정을 비디오 촬영한 후에 체크리스트를 사용하여 두 집단간의 차이를 통계적으로 분석하였다. 그 결과 과제수행, 과제분석, 계획하기 및 자기조절 능력에서 차이가 있음을 보여주었다[6].

정보과학 영재란 "일반적 지적 능력, 컴퓨터에 대한 강한 호기심, 높은 창의력, 수학-언어적 능력, 과제 집착력의 요소에서 모두 평균 이상의 특성을 소유하고 있는 사람 중에서 응용소프트웨어, 프로그래밍, 게임, 멀티미디어 등에 관심을 갖고 컴퓨터적 지각력, 일반화하는 능력, 추론력, 새로운 상황에 대처하는 능력, 문제를 분석하고 그들간의 관계를 파악하는 능력, 컴퓨터적 표현 능력, 적용력, 활용력이 뛰어나고 그 가능성이 있는 사람이다."고 정의하고 있다[13].

한편 정보과학 영재아의 특성에 관한 연구는 질적 연구 방법을 도입하는 경향을 뚜렷이 보여주고 있다. 면담 기법을 이용하여 IT영재의 특성을 추출하는 시도를 한 연구와 비디오 해석 기법을 이용한 중등 정보과학 영재아의 특성을 미시적으로 이해하려는 시도가 나타나고 있다[10][11]. 이러한 시도는 수학영재 또는 과학영재보다 상대적으로 최근에 등장한 정보과학 영재교육의 연구동향을 반영하듯 대한 실험적 연구로 진행되고 있으며 수집한 데이터로부터 영재아의 특성을 통합적으로 해석하려는 방식을 택하고 있다.

2.2 LOGO 프로그래밍 언어와 창의성

창의적 활동에서는 다양한 아이디어를 도출하는 브레인스토밍을 하는 확산적 사고를 통해 영재아의 흥미와 집중력 및 과제해결력을 신장시키게 된다. LOGO 계열의 컴퓨터 언어(MSWLogo, CricketLogo 및 NetLogo)는 이러한 창의적 활동

과 수학적 사고력을 신장하는 도구로서 각광받고 있다[15]. LOGO 언어는 Seymour Papert가 아동의 학습을 위해 고안한 프로그램 언어로서 학생들이 거북을 가르치는 과정에서 조직화된 체계적인 사고, 문제해결, 의사결정 등의 인지기술을 발달시키며, 수학을 가능하게 하여 수학 학습을 촉진시키는 수학적 환경을 제공한다[8][9][12][16].

Logo는 학습자 스스로 학습과 관련된 활동을 해 보면서 점점 아이디어를 발전시켜 나갈 수 있도록 해 준다. Logo는 절차적이고 재귀적인 언어이기 때문에 학생들로 하여금 자신의 생각을 구조화시키고 패턴을 분석하게 하여 여러 단계에서 작동시켜 본 명령어를 종합하게 한다. 또한 거북이를 대상으로 한 조작적 활동에 대하여 자신이 무엇을 하였는지를 생각하게 해 보거나, 조작적 활동의 결과를 돌이켜 보고 그것으로부터 수학적 의미를 추상화하는데 많은 도움이 된다[14][17].

초등학생들을 대상으로 실시한 MSWLogo 프로그래밍 학습에 관한 연구는 LOGO 프로그래밍 언어 학습 활동이 창의성 발달에 효과적인 것으로 보고하였다[15]. 구체적으로 보면 실험과 통제 집단으로 나누어 Torrance TTCT(도형) 사전-사후를 실시한 결과 LOGO 프로그래밍 언어가 세부항목인 유창성, 독창성, 제목의 추상성, 정교성 및 저항요인 향상에 전반에 걸쳐 창의성을 향상시키는 것으로 나타났다. 한편, 컴퓨터 예비교사를 대상으로 NetLogo 프로그래밍 활동을 탐구한 다른 연구는 정교성 요인을 제외한 유창성, 독창성, 제목의 추상성, 성급한 종결에의 저항 요인에서 실험 집단 학생들의 창의성이 향상되었음을 보여주었다[3].

2.3 계산적 사고와 프로그래밍에 관한 질적 연구

계산적 사고를 촉진시키기 위하여 하위 요소에 해당되는 분석적 사고, 논리적 추론, 알고리즘적 추론, 창의적 사고, 인지적 스킬 등을 다루는 교육과정이 필수적이다[5]. Scratch 프로그래밍 능력과 학습자 특성에 관한 연구는 수학, 과학, 정보교과에서 기본이 되는 논리수학적 지능은 계산적 사고를 다루는 문해교육(computational literacy)에서 정적 관계를 보여주는 결과를 제시하고 있다

[1]. 학습자의 특성을 살펴본 대목에서 연구자들은 적극적으로 정보를 다루고 탐색해보는 학습자들이 숙고형 학습자보다 프로그래밍 능력이 뛰어난 것을 보여주었다.

LOGO 프로그래밍을 하는 단계는 일반적으로 문제를 해결하는 과정과 비슷하기 때문에 컴퓨터 프로그래밍은 계산적 사고 기반의 문제해결의 한 유형이라고 종종 논의되어 왔다[12][14]. Polya가 주장한 문제해결의 네 단계(문제 이해, 계획 수립, 실행, 반성)는 아동이 프로그래밍 문제를 이해하고, 프로그램을 고안하고, 그것을 컴퓨터로 실행한 후 오류를 수정하는 것에 각각 대응한다고 볼 수 있다[16]. 이러한 단계에 따른 컴퓨터 프로그래밍의 과정을 비디오 촬영 및 면담으로 탐구한 질적 연구자들은 연구참여자 개인의 특성에 대한 부분을 심층적으로 다루고 있다.

초등학생 영재아들을 대상으로 진행한 연구는 LOGO 프로그래밍을 계획하는 과정과 오류 수정에 대하여 비디오 및 대화 자료를 분석하였다[12]. 영재아들은 기존의 지식과 절차를 활용하는 유추적 사고, 변수를 이용한 일반화, 여러 가지 명령어의 기능을 통합하여 활용하는 통합적 사고, 문제 해결을 위해 기존 명령어를 평가하는 비판적 사고, 현재의 상황을 새로운 관점에서 이해하고 응용하는 발전적 사고, 여러 가지 해결 방법을 구상하는 유연한 사고 등의 전략적 사고 유형을 보여주었다. 특히 그들은 Logo로 그림을 그리다가 오류가 나서 새로 그림을 그려야 할 때 부정적인 측면을 넘어서서 새로 작업함으로써 계획하지 않은 새로운 형태의 그림을 그리는 부수적 효과를 보여주었으며 이는 발전적이고 유연한 사고 활동에 긍정적인 영향을 주는 것으로 나타났다.

고등학생인 1명을 대상으로 수행한 질적 연구 사례에 등장하는 K는 처음에 하노이탑 문제를 천천히, 차근차근 읽어보고 직접 노트에 쓰기도 하고, 그 문제를 분석해서 패턴들을 찾아내고 생각을 구체화하여 알고리즘을 고안하였다. 그리고 그는 C 프로그래밍으로 자신이 고안한 방식을 코딩하고 테스트하였으며 에러를 수정하는 과정에서 ‘자기만의 생각을 실험해 보는 것에’ 재미를 느끼면서 분석적이고 절차적 사고를 증진시키는 사고 패턴을 보여주었다[4].

3. 연구설계

3.1 연구 대상자 및 수업 내용

이 논문에 등장하는 진우는 S대 과학영재교육원 수학IT융합반에 다니는 중학교 2학년 학생이다.¹⁾ 그는 초등학교 5학년 무렵에 6개월 정도 디키로 컴퓨터 프로그래밍을 처음 배웠으며 교사가 지시하는 대로 코딩해 보면서 다양한 음계의 소리를 내거나 LED를 사용하여 하트 모양의 빛이 나는 실습을 해 본 경험을 갖고 있었다. 그는 수학에 대하여 상당한 흥미를 느끼는 학생이었으며 수학 수업 외에 2012년 6월부터 9월 사이에 4회에 걸쳐 NetLogo 프로그래밍 수업을 받았다.

<표 1> NetLogo 수업 일정의 요약

수업 날짜	수업 내용 (3시간)
2012년 6월 9일	NetLogo의 간단한 명령어 소개 및 실습
2012년 6월 14일	어항 시뮬레이션
2012년 9월 1일	반복문을 사용하여 나선형 다각형 만들기, 재귀문을 사용하여 피보나치 수열 프로그램 짜기, 최대 공약수 구하기
2012년 9월 22일	별모양 만들기 (정다각형 만들기 및 다각별 만들기)

NetLogo 첫 수업에서 학생들은 기본적인 명령어(set if ifelse random 등)를 배우고 프로시저를 함수 형태로 만드는 것을 실습하였다. 어항 시뮬레이션 수업은 어항 속의 고기들의 움직임을 보여주고 한 마리의 물고기에 해당되는 터틀(turtle)의 움직임을 관찰할 수 있는 방법과 Behavior Space 사용법을 소개하였다. 반복문을 소개한 수업에서 강사는 시각적으로 이해할 수 있도록 나선형태의 삼각형의 모양을 반복적으로 그리는 예제를 통하여 코딩하는 방식을 제시하였으며 광역 변수와 지역변수의 사용법도 같이 소개하였다. 또한 학생들은 재귀문을 이용하여 피보나치 수열을 만드는 실습을 하였다. 별 모양 만들기 수업에서 강사는 정다각형을 만드는 예제를 소개한 후에 오각형의 별 모양을 만드는 방법을 다루었다. 그 이후에 그는 각의 수를 늘려나가는 프로젝트로

설정하고 각 학생들이 집에서 탐구할 수 있도록 과제를 내 주었다.

3.2 자료 수집 절차 및 분석

이 논문에서 연구자가 수집한 자료는 학생의 소감문, NetLogo 파일 (학생의 작업결과물), 면담, 비디오 자료이다. 비디오 기법은 교실 수업을 관찰하기 위하여 관찰 노트를 작성하는 방법은 면담 다음으로 많이 사용되는 질적 연구 방법에 속한다[10]. 면담 내용을 녹음한 오디오 테이프와 비교하면 비디오 테이프에 담긴 데이터의 양은 매우 방대하다. 비디오 테이프는 등장 인물이 서로 주고받는 대화뿐만 아니라 동작, 제스처 등의 화상 데이터까지 포함하고 있다. 이와 같이 비디오 촬영은 현장에 대한 기록을 충실히 저장하는 이점을 가지고 있으며, 반복하여 비디오 기록을 재생할 수 있는 점이 장점이다[12].

영재반 학생은 매번 3시간에 걸쳐 영재수업이 끝난 후에 학생이 그날 배운 핵심 내용과 자신의 소감을 적은 리포트 및 NetLogo 파일을 제출하였으므로 자료 수집이 용이하였다. 또한 연구자는 2차례의 면담(2013년 7월 30일과 2014년 1월 21일)과 비디오 촬영 (2013년 12월 14일)을 진행하였다. 연구자는 NetLogo 코드 분석과 비디오 해석을 통해 학생의 고유한 작업 패턴 또는 특성을 포착하고 그것이 가지는 의미를 연구자의 해석적 관점에서 풀어내었다[2][4].

연구자는 연구참여자와 사전 만남을 통하여 연구의 필요성과 목적 등을 공유함으로써 면담에 따른 연구참여자의 부담감을 덜어주었다. 사전에 구상한 질문은 영재반에 진학하게 된 동기, 수학에 대한 흥미, 컴퓨터 사용과 프로그래밍 경험 등에 관하여 만들어졌다.

연구자는 1차 면담에서 컴퓨터 프로그래밍 경험과 다각별을 NetLogo 언어로 코딩하는 과정에 대한 질문을 던졌다. 2차 면담에서 연구자는 수학과 컴퓨터 프로그래밍을 연계하면서 계산적 사고를 해나가는 양상에 대하여 깊이 있는 대화를 나누었다. 연구참여자가 실제 NetLogo를 실행하면서 설명하는 장면을 비디오로 녹화함으로써 연구자는 면담 자료를 해석할 때 보완 자료로 활용하였다.

연구참여자의 면담을 통하여 연구주제와 관련하여 필요하다고 판단되는 경우에 즉석에서 후속 질문을 던졌다. 면담 시간은 평균 1시간 정도였으며, 매 회 면담시마다 면담 내용을 녹음하였다. 그리고 그 면담한 내용을 전사 작업을 거쳐 전자 문서로 작성하였다.

면담 내용을 녹음기로 저장한 후에 전사를 하였으며 생생한 인용구 위주로 연구참여자의 고유한 특성을 드러내는 부분을 포착하였다. 계산적 사고와 컴퓨터 프로그래밍의 경험을 보다 생생하게 들려주는 오디오 데이터를 중심으로 주요 사건, 경험, 느낌 등을 시간 축 위에 배치하였고 생생한 인용구에 담긴 의미를 정교하게 드러냄으로써 내적 타당도를 확보하였다. 이러한 해석 작업을 통해 연구참여자 개인의 경험 속에서 지속적으로 고유하게 나타나는 학습자의 특성을 묘사하였다[4].

4. NetLogo 프로그래밍 수업 소개

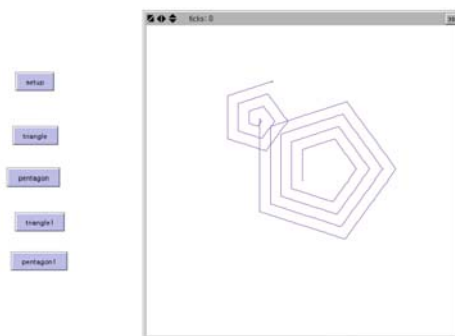
4.1 반복문으로 나선형의 다각형 그리기

학생들이 컴퓨터 프로그래밍 수업에서 초기에 배워야 할 내용은 반복적으로 계산하는 방식을 터득하는 것이다. 강사는 수업시간(2012년 9월 1일)에 약 1시간 30분에 걸쳐 반복문을 사용하여 프로시저 형태로 함수를 만들어서 실행하는 방법과 탐구 과제를 다루었다. 학생들은 정삼각형 등의 정다각형의 모양을 그리는 코드를 각자 해 본 후에 일반적으로 정점의 수가 n개 일 때 정다각형을 그리는 방식을 탐구하였다. 강사는 학생들과 대화를 나누면서 정다각형의 내각과 외각을 구하는 것이 중요함을 알아차리도록 안내하였다.

이어서 강사는 정다각형의 한변의 길이가 조금씩 증가하는 과제를 내 주었으며 정다각형의 한변의 길이가 늘어나는 현상에 착안하여 학생들이 변수를 사용하도록 자연스럽게 유도하였다. 진우는 정다각형의 성질을 쉽게 이해하였으며 내각과 외각의 관계를 수식으로 파악하여 아래와 같이 NetLogo로 코딩하였다.

```
globals [ a b ]
to setup ;; 거북이 그래픽 준비 단계
clear-all reset-ticks crt 1
ask turtles [ set heading 0 pd ] set a 5
end
to pentagon1 ;; 한 번에 오각형 그리기
let side 10
repeat 21 [ask turtles [ fd side rt 72]
set side side + 1]
end
to pentagon2 ;; 한 번에 한 변씩 그리기
ask turtles [ fd b rt 72 ]
set b b + 1
end
```

진우는 한 번에 오각형의 한 변이 그려지도록 하는 pentagon2 함수를 작성하였는데 이것은 계산의 절차가 한 번에 하나씩 실행되도록 함으로써 자신이 생각하고 있는 것이 제대로 구현되는지를 확인해 볼 수 있다. 이러한 절차적 방식을 화이트박스 기반 프로그래밍이라 부르며 계산적 사고를 컴퓨터 프로그래밍으로 전환시키는데 중요한 역할을 한다. 다시 말하면 학생이 단계별 계산의 수행을 파악하고 난 후에 한번에 전체 코드가 실행하도록 하는 방식으로 학습하는 것이 계산적 사고 기반의 프로그래밍 기능을 습득하는데 바람직하다.



<그림 1> 버튼을 이용하여 다양한 방식으로 오각형 그리기를 구현함

이러한 단계를 거친 학생은 일반적인 경우를 다룰 수 있는 코드를 간결하게 작성할 수 있는 단계에 도달하게 된다. 진우는 아래 코드에서 보는 바와 같이 네 개의 인수를 사용하여 일반적인

경우를 구현하였다. 즉 진우는 다각형의 개수(n), 반복 회수(m), 증분의 크기(l), 한 변의 크기(q)를 사용하여 원하는 모양을 화면에 나타낼 수 있도록 코딩하였다. 이러한 분석 결과를 볼 때 진우는 계산적 사고를 컴퓨터 프로그래밍으로 코딩하는 방식에 대하여 충실하게 이해하고 터득했음을 보여준다. 7각형 모양의 나선 형태를 만들기 위하여 진우는 “polyugon 7 10 4 3” 명령어를 실행하였다.

```
to polyugon [n m l q]
  let side q
  repeat m [ ask turtles [ fd side rt 360 / n ]
             set side side + l ]
end
```

4.2 재귀적 프로그래밍: 피보나치 수열

약 10분간의 휴식 시간을 가진 뒤에 강사는 재귀문을 사용한 NetLogo 프로그래밍 기법을 소개하였다. 피보나치 수열과 팩토리얼 계산 등에서 나타나는 연산의 패턴을 살펴보면 부분을 반복하여 나타나는 현상이 있다. 이를 재귀적 연산이라 부르는데 학생들은 이러한 패턴이 진행되는 방식으로 거의 유사하게 프로그래밍을 할 수 있다. 강사는 반복적으로 연산한다는 측면에서 재귀문을 사용하더라도 연산의 결과가 동일하다는 것을 설명하였다. 예를 들어 피보나치 수열을 계산하기 위하여 반복문을 사용한다면 두 개의 임시 변수(a와 b)를 사용해야 한다. 이를 NetLogo로 코딩하면 다음과 같이 변수를 다루는 약간의 테크닉이 필요하다.



<그림 2> polyugon 7 10 4 3 실행

```
to fibo2 [n]
  let a 1 let b 1 let temp 0
  type a type " " type b type " "
  repeat n [type a + b type " "
            set temp b
            set b a + b
            set a temp]
  print ""
end
```

강사는 이러한 반복문을 사용한 방식과 달리 재귀적인 연산을 NetLogo로 코딩하는 방법을 소개하였다. 1과 1과 출발하여 이전의 두 수를 합하여 새로운 수를 계산하는 부분을 아래 코드와 같이 처리하였다. 그런 후에 학생들은 print문을 사용하여 테스트하는 별도의 프로시저를 작성하였다.

```
to-report Fib [n]
  if n = 1 [report 1]
  if n = 2 [report 1]
  report (Fib (n - 1) + Fib (n - 2))
end
```

진우는 피보나치 수열을 계산하는 방식을 배운 대로 실습하면서 이해하는 것을 뒷받침하기 위하여 print문을 사용하여 연산의 중간 결과를 제시하는 방식을 나름대로 탐구하였다. 게다가 그는 연산을 기계적으로 수행하는 방식을 뛰어넘어 마치 닭이 알을 하나씩 낳듯이 피보나치 연산을 해나가는 방식을 탐구하여 그 과정을 시각적으로 보여주는 코딩을 해내는 놀라움을 보여주었다.

```
to setup
  clear-all create-turtles 1
  ask turtles [
    setxy random-pxcor random-pycor
    set size 1
    set shape "circle"
    set a 0]
end

to move
  ask turtles [
    set a a + 10
    if a >= 10[ hatch 1 [ set a -10 setxy
random-pxcor random-pycor] set shape "circle" ]
  ]
end
```

이러한 진우의 코딩 방식은 자신의 아이디어로 문제에 대한 단서를 찾고 이를 하나씩 계산해 나가는 절차적 프로그래밍으로 변환시킴으로써 계산적 사고에 바탕을 둔 프로그래밍 실력이 부족해져나오는 결과를 보여주었다. 흥미로운 부분은 그가 피보나치 수열을 재귀문 전용 문법으로 연습한 후에 자신만의 아이디어를 내어서 hatch와 광역변수 등을 사용하여 단계별로 버튼을 눌러서 피보나치 수가 한 개씩 생성되는 과정을 알 수 있는 형태로 만들어 보았다는 점이다.

수업 끝 무렵에 학생들은 각자 재귀문을 사용하여 두 수의 최대공약수를 찾는 방법을 코딩해보다가 수업이 종료되었다. 수업 후에 진우는 회상해 보면서 재귀문이란 “반복을 하되 1회 반복할 때 마다 변화를 주는 식의 반복”으로 인지하고 있었음을 보여주었다. 그리고 그는 “globals [a]”라고 코딩하면 a가 광역변수로 존재하지만 각 명령어 옆에 [매개]변수를 지정하는 방식을 사용하여 그 변수가 그 구간에서만 존재하게 하는 방법도 알게 되었다.

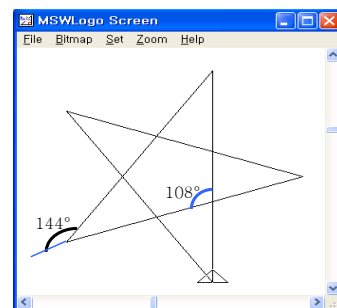
진우는 자신이 생각하는 문제에 대한 답을 찾아가는 탐구 방식이 NetLogo 프로그래밍을 도구로 하여 명료하게 구현하고 있었다. 이러한 그의 활동은 그가 작성한 소감문에서 NetLogo 프로그래밍을 통하여 새로운 문제에 도전하고자 하는 도전감과 성취감을 느끼고 있음을 생생하게 반영하고 있다.

진우: “나는 시간만 더 있었으면 질리지 않고, 재귀문을 이용해서 최대공약수를 구하는 방법을 탐구하고 설명을 들으면서 내가 생각한 방법과 어떻게 다른지 알아보고 싶었다. 하지만 시간이 부족했고, 다음에라도 선생님이 기억하시면 이것을 한 다음에 다음수업으로 들어가고 싶다. 넷로고를 이용해서 별의 별꼴 다 한다는 생각이 들었고, 넷 로고는 한번 프로그램을 짜놓으면 정말 편하지만 프로그램을 짜는 것이 쉽지 않고, 다양한 방면에서 쓰일 수 있겠다는 생각을 하였다.”

4.3 NetLogo 프로그래밍: 다각별 만들기

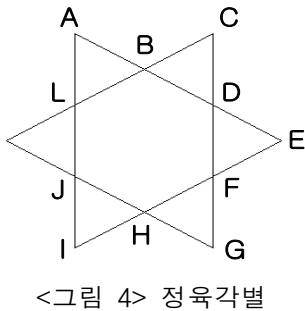
학생들은 이전 수업시간에서 이미 NetLogo의 기본 명령어를 사용하여 하나의 프로시저(함수)로 모든 정다각형을 그릴 수 있었다. 그런 다음 강사는 별 모양을 만들어보기로 했는데 오각형에서 출발하여 각의 수를 늘려나가는 경우를 다루도록 안내하였다. 그러나 학생들은 그것이 쉽지 않음을 알아차렸고 수업시간 안에 하지 못하여 숙제로 내주었다.

강사는 먼저 [그림 3]과 같은 별을 소개한 후 학생들은 삼각형 외각의 개념을 활용해서 별모양을 쉽게 그릴 수 있었다. 여기서 NetLogo 수업이 수학적 개념을 바탕으로 이루어지고 있음을 알 수 있다. 강사는 영재 학생들의 흥미와 탐구력을 향상시킬 목적으로 다음과 같이 학생들에게 제안하였다. 먼저 그는 학생들과 약간의 의견 교환을 한 다음에 꼭지점의 개수가 6개, 7개, 8개 등 늘어나는 경우에 별 모양을 만들어보도록 격려했다. 다음의 그림은 내부의 정오각형과 삼각형의 외각을 이용해 만들어진 별 모양이다.



<그림 3> 별 모양 그리기

약 30분이 지나자 몇몇 학생들은 홀수개의 다각별 모양을 그릴 수 있으나 짝수개의 경우에 문제가 생긴다는 것을 발견하였다. 학생들은 이 문제가 쉽게 풀리지 않음을 알아차리고 다른 방법을 모색하였지만 별다른 아이디어가 떠오르지 않고 끁끁대기만 하였다. 이에 교사는 별 테두리와 내부 다각형을 그리도록 구분하였고 다음과 같이 힌트를 주었다. 그런 후에 개인별 집에서 틈나는 대로 프로그래밍을 해 보도록 과제로 내 주었다. 약간의 동기 부여를 위하여 NetLogo 프로그램을 완성하여 이메일로 제출하는 학생들에게 선물을 준다고 공지하였다.



<그림 4> 정육각별

진우는 1주일 가량 지난 후에 다음과 같은 코드를 이메일로 제출하였다. 코드를 분석해 본 결과 내부다각형을 먼저 그리고 이어서 외부테두리를 그리는 두 부분으로 구성되어 있음을 알 수 있었다. 진우는 먼저 외각을 사용하여 다각별 내부에 해당하는 정다각형을 쉽게 그려나갈 수 있었다. 이어서 그는 정이등변삼각형의 성질을 이용하여 세 각의 크기와 삼각함수를 사용하여 빗변의 길이를 계산하여 외부테두리를 그려나갔다.

```

to star
  ;; 내부다각형 그리기
  repeat num_p [ rt 360 / num_p fd 10 ]
  ;; 외부테두리 그리기
  repeat num_p [ fd 5 / cos (360 / num_p) rt 720 / num_p fd 5 / cos (360 / num_p) lt 360 / num_p ]
end
    
```

<코드 1> 진우의 다각별 그리기 NetLogo 코드

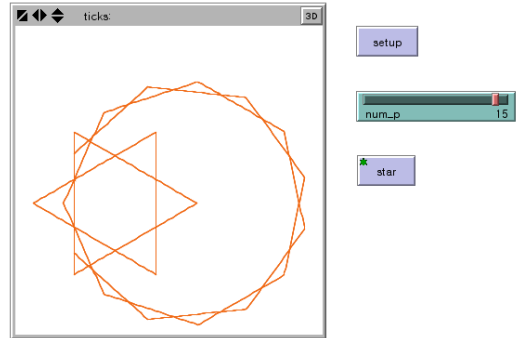
여기서 num_p는 슬라이더의 값이고 두 번째 줄은 먼저 num_p각형을 그리는 것이고 세 번째 줄은 바깥쪽 삐죽삐죽 튀어나온 부분을 그리는 것이다. 예를 정육각형의 경우에 다음과 같이 실행된다.

(내부 정다각형 그리기)

LB → BD → DF → FH → HJ → JL

(외부 테두리 그리기)

→ LA → AB → BC → CD → DE → EF
 → FG → GH → HI → IJ → JK → KL



<그림 5> 정6각별과 정15각별 모양

이 코드는 정확하게 두 개의 모듈로 구성된 아이디어를 명료하게 보여주고 있으며 코딩 역시 간결하게 작성되어 있다. 게다가 거북이의 출발점(L)을 보면 위로 올라가지 않고 바로 오른쪽으로 회전한 후에 직진을 함으로써 내부 다각형을 그린 후에 그 지점(L)에서 바로 LA 외부 테두리를 그려나감으로써 군더더기가 없이 깔끔한 코딩 실력을 보여주었다.

그는 어떻게 삼각함수의 코사인을 사용할 생각을 하게 되었을까? 연구자는 두 차례의 면담에서 이를 탐구하였다.

(2013년 7월 30일 면담)

연구자: 이 문제 처음에 과제로 나왔을 때 느낌이 어땠어요?

진우: 처음에는 쉬울 줄 알았어요. 그냥 이런 식으로 그리면 될 줄 알았는데 짝수 각형에서는 이게 안 통하는 거예요. 예를 들어가자고 육각형 같은 경우는 모양이 이런 식으로 되잖아요. 이거를 이런 식으로 그릴 수가 없었어요. 어떻게 해도 안되요 이게 잘. 그래가지고 생각해 낸게 이거

연구자: 그거를 어떻게 생각해 냈는데. 막 몇날 며칠을 고민했어요?

진우: 아니요.

연구자: 그 정도는 아니었고

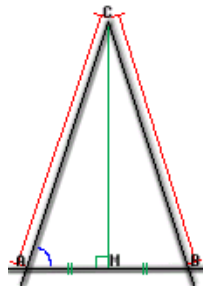
진우: 네 학교에서 번뜩하고 생각났어요.

그는 다각별 그리기에서 중요한 아이디어를 지속적으로 찾다가 삼각형의 성질에 관한 정보(내각과 외각)와 삼각함수에 대하여 생각이 미치자 순간적으로 해법을 찾게 되었음을 확인할 수 있다. 이것은 그가 손에 넣은 문제에 대하여 생각의 끈

을 놓지 않고 문제해결의 단서를 찾아가는 특성을 보여주고 있다. 정다각형의 내부 모양과 외부 테두리를 그려나가는 방식에 대한 아이디어를 설명하는 부분도 매우 간결하다.

연구자: 학교에서 계속 생각하고 있었어요? 그 문제를 어떻게 풀 것인가?

진우: 아니요. 계속 생각하고 있었던 거 아닌데 지내다가 갑자기 그 생각을 했는데 번뜩하고 생각났어요. (중략) 적고 나서 이렇게 직각삼각형을 그려가지고 이제 계산을 했어요 (중략) 다각형이 있으면 이렇게 되잖아요. 그러면 정다각형이기 때문에 몇 각형인지만 알면은 여기 각[정다각형의 내각]도 알 수 있고 여기 각[정다각형의 외각]도 각도를 알 수 있고 또 [삼각함수를 사용하여] 길이의 비를 알 수 있어요. 그거를 이용해 가지고 전부다.... 그런 식으로 했어요.



<그림 6> 이등변삼각형의 밑변의 길이

<그림 6>에서 선분 CM은 이등변삼각형 꼭지점 C에서 밑변 AB로 뺀 수선으로 이등변삼각형의 성질에 의해 밑변 AB를 수직이등분한다. 직각삼각형 AMC의 선분 AM 또는 빗변 AC의 길이는 아래와 같은 수식으로 구할 수 있다.

$$\cos \quad CAM = \frac{AM_r}{AC_r}$$

연구자는 그가 평소에 수학을 할 때 어떤 방식으로 문제를 해결하는지 후속 면담(2014년 1월 21일 면담)에서 탐구하였다.

연구자: 그림을 그리고 하여튼 아이디어를 캐치한 다음에 그 아이디어를 캐치해서 쓰는데 시간은 대략 얼마나 걸려요?

진우: 음.. 대략... 그림을 그리고 나서 문제를

읽은 순간부터 그림을 머릿속에 정리를 한 다음에 이게 안 풀릴 것 같다 해서 그림을 그리고.. 생각을 정리 해가지고 아이디어를 딱 생각하면은 딱 정리 다 풀 때가지요?

연구자: 음 일단 아이디어에서 쓰기까지

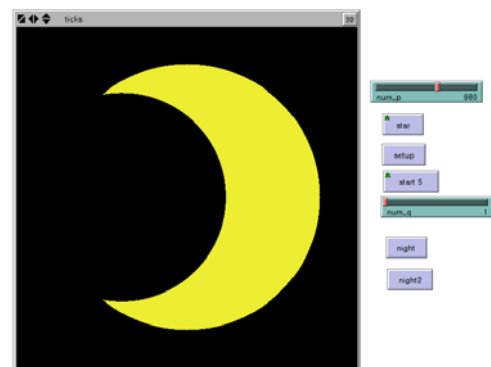
진우: 쓰기까지요? 글쓰는 시점 까지요?

연구자: 음

진우: 아마 한....시간대가 다를 것 같은데 30초에서 한 2분까지요

진우는 평소에 문제를 대할 때 먼저 읽어보고 머리 속에서 문제를 풀 아이디어를 구상하면서 그림을 그리기도 하였다. 그는 평소에 차분히 숙고하면서 머리 속에서 정리한 아이디어를 토대로 답을 찾거나 NetLogo 코딩을 하는 패턴을 보여주었다.

그는 이에 그치지 않고 자신이 만든 다각별 전체를 모듈화하여 하나의 다각별 속에 다른 다각별을 합쳐서 보임에 함으로써 그만의 초승달 창작품을 만들기도 하였다. 그는 다각별 코드를 사용하여 재귀문으로 원 모양의 안을 채우는 방식을 택하였으며 이차부등식을 사용하여 원 밖의 영역을 까만 색으로 칠하여 밤을 나타내었다.



<그림 7> 정980각형을 사용한 초승달

```
to start [ n ]
  if n > num_p + 1 [ stop ]
  repeat n [ rt 360 / n fd 1 ]
  repeat n [ fd 0.5 / cos (360 / n) rt 720 / n
fd 0.5 / cos (360 / n) lt 360 / n ]
  start ( n + num_q )
end
```

to night

```
ask patches [if pxcor ^ 2 + pycor ^ 2 >
64000 [set pcolor black]]
end
```

4.4 다각별 LOGO 수업 사례간 고찰

여러 가지의 아이디어를 모색하고 걸러내는 작업을 거쳐 정제된 컴퓨터 프로그래밍으로 구현하는 진우의 사례를 보면 마치 숙련된 문제해결자의 모습을 볼 수 있다. 여기서 연구자는 다각별 그리기 수업에 등장하는 다른 연구 사례와 비교해서 영재학생의 특성을 고찰해 보고자 한다. 비교 대상은 2004년 1월 15일에 진행된 별 그리기 수업(총 5시간)을 촬영한 비디오 자료에서 등장하는 남학생 지훈과 명석의 사례이다.

전반적으로 강사는 다각별 모양 그리기 활동에 관한 영재 수업에서 학생들에게 먼저 아이디어를 묻고, 학생들이 각자의 의견을 발표하면서 서로의 생각을 이해하고, 더 좋은 아이디어와 방법을 찾아가도록 안내하였다. 이러한 수업은 브레인스토밍을 통하여 여러 가지 아이디어를 모색하는 발산적 사고를 촉진시키면서 아이디어를 발전시켜 해답에 근접해 가는 수렴적 사고로 연계되는 수업의 특징을 보여주었다[10].

대부분의 학생들이 어렵다고 생각한 아이디어를 지훈이는 계속적으로 탐구하여 그 자신만의 방법을 찾아나갔다. 처음에 지훈이가 생각한 방법은 별 가장자리에 외부 테두리를 채워가는 방식을 고안하였다. 지훈이는 자신의 생각을 다음과 같이 MSWLogo로 코딩하였다.

```
to star :n :a
repeat 3 [fd :a rt 120] ;; 정삼각형 만들기
rt 60 fd :a ;; 한변의 길이만큼 가기
rt 360/:n ;; 오른쪽으로 회전
repeat :n-1 ~
[repeat 3 [fd :a lt 120] fd :a rt 360/:n]
end
```

<코드 2> 지훈이의 다각별 그리기 MSWLogo 코드

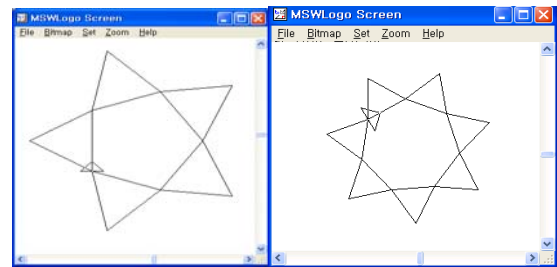
그런데 지훈이가 작성했던 코드는 조금 더 생각해서 정리해 보면 다음과 같이 간결하게 바꿀

수 있다. 여기서 우리는 그가 꼭지점의 수에 관계 없이 다각별을 그리는 해법에 근접함을 볼 수 있다.

```
to star2 :n :a
repeat :n [repeat 3 [fd :a lt 120] fd :a rt 360/:n]
end
```

<코드 3> 지훈이의 MSWLogo 코드를 간소화함

<그림 8>은 지훈이가 자신의 아이디어를 MSW Logo로 구현해 본 정오각별 그림이다.



<그림 8> star 5 50(좌)와 star 7 50 (우)

이와 같이 지훈이의 아이디어대로 별 모양은 완성되었으며 원하는 별모양에 근접한 결과를 얻게 되었다. 그러나 그는 변의 모양이 불룩하게 나오는 모양(좌)과 오목하게 들어간 모양(우)을 교정하기 위하여 하나의 꼭지점과 다른 꼭지점을 잇은 선분이 직선이 되도록 하는 다른 아이디어를 찾아보기로 하였다.

약 15분 정도의 브레인스토밍을 거쳐도 아이디어가 나타나지 않자 강사는 직각삼각형을 이용한 힌트를 주었으며 삼각비의 개념을 도입하였다. 그는 삼각함수의 COS 개념을 설명해 주고 휴식 시간을 취하였는데 학생들은 쉬는 시간 동안 인터넷 검색을 통해 삼각비에 대한 내용을 조금 더 분명하게 이해할 수 있었다. 그러나 지훈이는 다각별의 모양을 정확하게 그려내지 못하였다.

반면에 명석은 차분하게 하나씩 코드를 실행해 보면서 이와 같은 방법을 이용해 외부 테두리를 그리고 나서 내부의 다각형을 그려나갔다. 여기서 코사인을 사용하여 직각삼각형의 빗변의 길이를 구하였고 MSWLogo 언어로 코딩하여 짝수 홀수에 관계없이 별 모양을 그릴 수 있었다. 다음은 명석이가 작성했던 일반적으로 다각별 모양을 그리는 코드이다.

```

to nstar :n :a
  ;; 외부 테두리 그리기
  repeat :n [fd :a rt 360/:n fd :a lt 720/:n]
  fd :a
  ;; 내부 다각형 그리기
  repeat :n [fd 2*(cos 360/:n)*:a lt 360/:n]
end
    
```

<코드 4> 명석이의 다각별 만들기 MSWLogo 코드

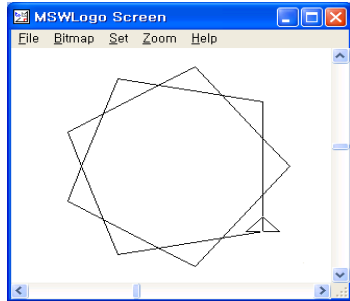
위의 코드를 자세히 분석해 보면 다음과 같은 순서대로 <그림 4>의 육각별을 그렸음을 알 수 있다.

(외부 테두리 그리기)

→ FE → ED → DC → CB → BA → AL
 → LK → KJ → JI → IH → HG → GF

(내부 정다각형 그리기)

FD → DB → BL → LJ → JH → HF



<그림 9> 꼭지점 9개인 별: nstar 9 50

4.5 계산적 사고에 관한 사례간 비교 고찰

진우는 정육각별에 관한 아이디어를 머리 속에서 탐색하면서 필요한 경우에 그림을 그려서 아이디어를 발전시키곤 하였다. 짝수 개의 꼭지점을 가진 정다각별을 그리기 위하여 그는 학교 생활에서도 지속적으로 문제해결의 단서를 찾는 끈질김을 보여주었고 순간적으로 포착한 아이디어를 토대로 NetLogo로 계산하는 작업 패턴을 보여주었다. 프로그래밍 경험이 부족함에도 그는 이러한 수학적 사고력을 컴퓨터 계산으로 풀어내는 방식을 명료하게 파악하고 있었으며 [코드 1]에서 보듯이 NetLogo의 명령어를 간결하게 사용하여 코

딩을 하는 과정을 보여주었다. 그는 정다각별을 그리는 과제를 두 개의 모듈로 나누고 복잡한 그리기 방식을 간단하게 단순화 시키면서 NetLogo로 코딩하는 추상화와 자동화의 측면을 보여주었다.

명석은 스스로 수학적 아이디어를 탐색하고, 주어진 과제보다 더 많은 것을 처리하며, 다른 학생들에 비해 심사숙고하는 능력과 강한 집중력을 보여 주었다. 그는 수업시간에 교사의 말을 정확히 이해하였고, 그것을 자신의 것으로 만들어가는 경향을 보여주었다. 강사가 어떤 문제에 대해 질문을 하면 그는 대부분 논리적으로 상세하게 대답하였다. 그는 또한 자신이 탐구한 삼각함수의 기능에 관하여 이해를 분명히 하였다. 다각별과 같이 개방형 문제를 해결하는 과정에서 명석은 중요한 원리를 파악하고, 관계를 파악하는 능력, 원인과 결과에 대한 정확한 예측을 하는 등 정보 처리 과정을 통해 문제를 해결하는 정형화된 사고의 틀을 보여주었다.

지훈은 상대적으로 그런 특성이 약하게 나타났지만 자신의 아이디어를 MSWLogo 프로그래밍을 통해 실험해보고 그 결과를 보면서 아이디어를 적극적으로 탐색해 나가는 패턴을 보여주었다. 그는 다각별 그리기에서 정삼각형을 연속적으로 그려나가면서 내부에 정다각형을 만들어 가는 독창적인 아이디어를 제안하는 등 브레인스토밍을 사용한 적극적인 사고를 하는 경향을 보여주었다. 즉, 그는 도달하고자 하는 목표 상태에 이르는 절차를 자신만의 방법으로 고안하여 그리는 방식을 계산함으로써 목표 상태에 근접하는 알고리즘을 보여주었다. 지훈은 다른 학생들이 어렵다고 생각한 아이디어를 계속적으로 탐구하여 자신만의 방법으로 다양한 방법을 탐구하는 계산적 사고의 경향을 보여주었다.

5. 논의 및 제언

본 연구는 정보과학 영재반 학생들을 대상으로 수학과 정보를 사용하여 LOGO 컴퓨터 프로그래밍을 하면서 정다각형의 변형인 다각별 문제를 해결하는 활동과 창의적 활동의 양상을 질적 사례 연구로 다루었다. 세 명의 연구참여자가 보여

준 LOGO 학습 활동에 관하여 사례 형태로 고찰해 본다.

5.1 창의성에 관한 논의

다각별 모양 그리기는 수학적 이론을 바탕으로 이루어지고 있는데, 삼각형 외각의 개념에서 이등변 삼각형, 삼각함수까지 폭 넓게 적용되었다. 진우는 NetLogo의 기본 프로그래밍에서 프로시저를 사용하고, 변수를 정의 하는 등 최소한의 프로그래밍을 위한 기초만 갖추었음에도 자신의 수학적 능력을 토대로 다각별의 홀수 경우에 대한 해법을 순식간에 찾아내는 경험을 하였으며 이를 NetLogo 언어로 간결하게 코딩하는 실력을 보여주었다. 그가 학교에서 다각별의 해법에 대하여 생각하다가 순간적으로 아이디어를 포착한 경험은 과학자들이 특정 문제를 해결하기 위하여 골똘히 생각하다가 일상 생활에서 홀연히 해결의 실마리를 얻게 되는 경험과 유사한 점이 있다. 그가 작성한 코드에 대한 분석 결과는 그가 수학적으로 생각하는 바를 절차적인 알고리즘을 통해 컴퓨터 언어로 구현해 봄으로써 수학적 생각과 계산적 절차가 매우 긴밀하게 연결되어 있음을 시사하고 있다.

반면에 명석은 다각별과 같이 개방형 문제를 해결하는 과정에서 중요한 원리를 파악하고, 관계를 파악하는 등 집중력을 보여주었다. 그리고 지훈은 다각별 문제 해결 과정에서 자신만의 절차를 모색하는 등 독창적인 아이디어를 탐색하는 측면을 보여주었다.

사례간 비교고찰 결과 진우, 명석, 지훈 세 명의 영재반 학생들은 LOGO 프로그래밍을 통해 계산적 사고를 바탕으로 문제를 해결하였으며 이러한 과정은 창의성 연구에서 제시된 특징에 근접하고 있었다[3][7]. 즉 그들은 다각별 모양 그리기 과제를 통하여 다양한 관점에서 특정한 문제를 해결하려는 융통성과 특정한 아이디어에 구체적인 사항을 덧붙이는 정교성을 지향하면서 더 이상 진전이 없을 때 지속적으로 아이디어를 탐색하는 과제집착력을 보여주었다[6]. 그들은 새로운 아이디어에 대한 단서를 포착하거나 문제해결에 대한 도움을 받았을 때 문제해결과정에서의 곤란

한 점을 파악하여 빠트린 정보를 탐색할 수 있는 문제에 대한 민감성과 프로그래밍 오류 발생시에 세부사항을 분석하고 전체를 부분으로 분석할 수 있는 능력 등을 보여주었다.

5.2 기존 연구와 비교 고찰

이 연구는 LOGO 프로그래밍을 통하여 영재아들이 문제해결의 과정에서 창의적 활동과 계산적 사고를 활성화 시키는데 관련된 질적 사례를 제공하고 있다. 관련된 연구로는 초등학교 6학년 영재반 학생들이 MSWLogo를 사용하여 다각별 만들기를 하면서 수의 관계성(배수, 약수, 소수 등)을 수학적으로 탐구한 논문이 있다[14]. 영재수업용 교재를 개발하는 쪽에 비중을 두었던 것에 반하여 이 연구는 참여 학생들의 사고 및 프로그래밍 작업 패턴, 학습태도, 행동특성 등을 분석하였다. 비디오 촬영과 면담 자료를 토대로 피보나치 수열 생성과 다각별 모양 그리기 등에 관한 LOGO 프로그래밍 활동에 대한 분석을 통하여 계산적 사고를 전개시켜 나가는 학생 개인의 특성과 창의적 활동의 발현에 관한 자료를 미시적으로 해석하는 토대를 제공했다는 점에서 기여하는 바가 크다.

구체적으로 보면 삼각함수를 사용해야만 해결할 수 있는 다각별을 다루는 사례에서 중학생들은 초기의 아이디어를 포착하여 발전시켜 나가는 양상과 LOGO 언어로 코딩하는 과정에서 보여준 개인의 특성이 사뭇 다르게 나타나고 있음을 볼 수 있었다. 명석과 지훈이는 변수를 이용한 일반화 및 여러 가지 명령어의 기능을 통합하여 활용하는 통합적 사고, 현재의 상황을 새로운 관점에서 이해하고 응용하는 발전적 사고, 여러 가지 해결 방법을 구상하는 유연한 사고 등을 보여줌으로써 초등 수학반 영재학생들을 대상으로 진행한 질적 연구에서 다양한 전략적 사고가 나타남을 볼 수 있다[12].

반면에 진우는 문제를 이해하고 해를 찾아가는 과정 속에 분석적이고 추론적인 사고가 연관되어 나타나며 [코드 1]를 보면 수학적 계산을 토대로 컴퓨터 프로그래밍을 논리적으로 구현하는 전문가적인 특성을 보여주었다. 또한 <그림 7>에서

보는 바와 같이 정980각형을 사용하여 초승달을 그려볼 정도로 유연한 사고를 하고 있음을 볼 수 있다. 이처럼 진우는 양적 연구에서 나타나는 창의성의 하위 요소 중에서 자신의 아이디어를 가지고 실험적으로 다양하게 발전시켜 나가는 유연성을 보여주었다[3][15]. 이것은 다양한 아이디어의 모색과 회소성을 감안할 때 유창성과 독창성과 관련이 되는 것으로 보인다. 또한 더 이상 진전이 없을 때 그는 성급하게 종결하지 않고 나온 아이디어에 대한 지속적인 탐색을 함으로써 순식간에 해결의 실마리를 포착하는 양상을 질적 사례로 보여주었다.

5.3 시사점과 제언

이 연구는 참여 학생들의 수가 세 명인 점을 고려할 때 계산적 사고를 토대로 창의적 활동을 보여주는 질적 연구의 결과를 일반화 하는데 한계가 있다. 하지만 흥미로운 부분은 LOGO 프로그래밍을 지속적으로 하는 과정에서 나타났다. 그들은 단기간에 특정 능력이 향상되는 기미를 보이지 않다가도 6개월 또는 1년 이상 장기간에 걸쳐 LOGO 프로젝트 활동을 하다보면 특정 시점이 지나면서 창의성 또는 계산적 사고력의 향상을 가져 올 수 있음을 보여주었다. 프로그래밍 초보자였던 진우는 자신의 수학적 사고력 즉 계산적 사고를 토대로 거꾸로 문제해결에 필요한 NetLogo 프로그래밍의 명령어를 습득하는 일종의 후향식 추론과 같은 프로그래밍 학습 패턴을 보여주었다.

이러한 연구결과는 정보과학 영재수업에 창의성과 계산적 사고력을 증진하기 위하여 개선해야 할 방향을 보여준다. 첫째, 수학적 사고력과 알고리즘적 사고를 신장시키기 위하여 아이디어 찾기, 아이디어 발전시키기 등의 브레인스토밍 과정에서 문제 해결의 단서를 생각하고(Thinking) 그것을 말하면서(Speaking) 표현하며 그런 다음에 쓰면서(Writing) 정리해 보는 학습 방법에 대한 탐구가 필요하다. 이런 과정을 거쳐 계산적 절차를 코딩과 오류 수정을 통한 문제해결의 과정을 촉진시키는 컴퓨터 프로그래밍 수업을 적극 도입해야 한다[2][9]. 둘째, LOGO 언어로 코딩하는 과

정에서 원하는 결과가 나오지 않았을 때 원인을 논리적으로 분석하고 발표하면서 나선 형태로 다시 코딩 및 오류 수정을 하는 교수-학습 방법을 적극 도입한다[4].

후속 연구는 학습자의 특성에 따른 계산적 사고와 창의성의 관련성을 살펴보는 것이다. 예를 들어, 적극적으로 정보를 다루고 탐색해보는 학생들(지훈)과 가만히 앉아서 어떤 아이디어가 떠오를 때까지 기다리면서 숙고하는 학생(진우와 명석)들에 대하여 질적 사례 연구를 깊이 있게 진행할 필요가 있다. 이러한 대비는 질적 자료를 수집할 수록 그 차이가 상당한 대비를 이루고 있음을 보여주지만 양적 연구로 포착하기에는 한계가 있다고 판단된다. 마찬가지로 맥락에서 LOGO 프로그래밍 활동을 하는 학생들이 TTCT 검사지에서 분류한 창의성의 발현 과정에 따라 유창성, 독창성, 추상성, 성급한 종결에의 저항 요인 등 미시적으로 나타나는 질적 자료를 통합적으로 해석하는 것은 후속 연구로 남긴다[3][4].

후 주

1) 연구참여자의 개인 정보보호 차원에서 익명을 사용하였다.

참 고 문 헌

- [1] 김수환, 한선관, 김현철 (2010). Computational Literacy 교육에서 프로그래밍 능력과 학습자 특성에 관한 연구 -학습스타일과 다중지능을 중심으로-. **컴퓨터교육학회논문지**, 13(2), 15-23.
- [2] 김수환, 한선관, 김현철 (2011). 프로그래밍 과정에서 나타나는 초보학습자들의 행동 및 사고 과정 분석. **한국컴퓨터교육학회 논문지**, 14(1), 13-21.
- [3] 김진영, 박홍준, 전영국 (2011). PBL 기반의 NetLogo를 이용한 교육용 콘텐츠 저작이 예비 중등교사의 창의성에 미치는 효과. **공학교육연구**, 14(4), 29-38.
- [4] 장정숙, 전영국, 윤지현 (2013). 하노이 탑 프로그래밍 경험에서 나타나는 정보과학적 사고 패턴에 관한 질적 사례 연구. **한국컴퓨터교육**

학회 논문지, 16(4), 33-45.

- [5] 최숙영 (2011). 21st Century Skills와 Computational Thinking 관점에서의 ‘정보’ 교육과정 분석. **컴퓨터교육학회논문지**, 14(6), 19-29.
- [6] 이혜주 (1994). **영재와 일반아동의 문제해결 과정 비교 연구**. 동덕여자대학교 석사 논문.
- [7] 김종훈, 김종진, 정원희 (2005). 프로그램 요소를 이용한 창의성 신장 교재 개발 연구. **한국컴퓨터교육학회 논문지**, 8(5), 17-30.
- [8] 이현석, 이수정 (2009). 초등학생의 프로그래밍 능력과 특기적성간의 관계 분석-로고 언어 활용 사례. **한국컴퓨터교육학회 논문지**, 12(3), 23-30.
- [9] 전영국, 김진영, 김정섭, 조인성, 허영현 (2006). **머리가 좋아지는 LOGO 프로그래밍**. 서울: 교우사.
- [10] 윤지현 (2004). **비디오 분석기법을 이용한 정보과학 영재 학생의 질적 사례 연구**. 순천대학교 교육대학원 석사 학위 논문.
- [11] 주수희 (2006). **프로그래밍 교육을 통한 정보과학 영재의 질적 사례 연구**. 순천대학교 교육대학원 석사학위 논문.
- [12] 류희찬, 장인욱 (2010). LOGO를 이용한 프로젝트 학습에서 나타난 초등 수학영재 학생들의 전략적 사고. **대한수학교육학회지 수학교육학연구**, 20(4), 459-476.
- [13] 김미숙, 이재호 (2005). **정보과학 영재교육을 위한 교육과정, 선발도구 및 교수학습자료 개발**. 한국교육개발원 수탁연구 CR2005-52.
- [14] 이은석 (2013). **LOGO에서 별다각형 그리기를 활용한 초등수학영재교육 학습 지도 자료 개발**. 공주교육대학교 교육대학원 석사학위 논문.
- [15] 이점순 (2008). **LOGO 프로그래밍 언어가 초등학생의 창의성 발달에 미치는 영향**. 전주교육대학교 교육대학원 석사학위 논문.
- [16] Abelson, H. & diSessa, A. (1988). *Turtle Geometry*. Cambridge: The MIT Press.
- [17] Harvey, B. (1985). *Computer Science Logo Style: Symbolic Computing*. Cambridge: The MIT Press.

전영국



1986 수원대학교 수학과 (이학사)

1986 시카고주립대학교 수학과(이학석사)

1995 일리노이대학교 어바나-삼페인(교육학박사)

1996~현재 순천대학교 사범대학 컴퓨터교육과 교수

관심분야: 튜터링 시스템, 로봇과 예술의 융합, 에이전트 기반 프로그래밍 등

E-Mail: ycjun@sunchon.ac.kr