

융합 수업 프로그램에서 나타나는 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고 분석¹⁾

An Analysis on the Mathematical Creativity and Computational Thinking of Elementary School Mathematical Gifted Students in the Convergence Class Programs

강 주 영 · 김 동 화²⁾ · 서 혜 애

ABSTRACT. The purpose of this study is to analyze the mathematical creativity and computational thinking of mathematically gifted elementary students through a convergence class using programming and to identify what it means to provide the convergence class using Python for the mathematical creativity and computational thinking of mathematically gifted elementary students. To this end, the content of the nine sessions of the Python-applied convergence programs were developed, exploratory and heuristic case study was conducted to observe and analyze the mathematical creativity and computational thinking of mathematically gifted elementary students. The subject of this study was a single group of sixteen students from the mathematics and science gifted class, and the content of the nine sessions of the Python convergence class was recorded on their tablets. Additional data was collected through audio recording, observation. In fact, in order to solve a given problem creatively, students not only naturally organized and formalized existing mathematical concepts, mathematical symbols, and programming instructions, but also showed divergent thinking to solve problems flexibly from various perspectives. In addition, students experienced abstraction, iterative thinking, and critical thinking through activities to remove unnecessary elements, extract key elements, analyze mathematical concepts, and decompose problems into small components, and math gifted students showed a sense of achievement and challenge.

Received July 6, 2022; Revised August 3; Accepted August 26, 2022.

2010 Mathematics Subject Classification : 97D40

Key Words : Mathematical Creativity, Computational Thinking, Gifted Education

1) 이 논문은 강주영의 2022년 박사학위논문을 바탕으로 수정·보완하였음.

2) Corresponding Author

I. 서론

2016년 스위스 다보스에서 열린 세계경제포럼에서는 ‘4차 산업혁명’이라는 주제로 10년 이내 급격한 사회적 변화를 예견했다. 4차 산업에 관한 관심의 증가와 인공지능(Artificial Intelligence, AI)의 발달과 함께 정부에서는 사회·경제 전반의 혁신을 위해 소프트웨어 교육과 인공지능 역량을 강조하고 있다. 이러한 시대적 흐름을 바탕으로 영재교육도 많은 연구자들이 4차 산업혁명 시대에 대응해 다양하게 노력하고 있다. 그 중 소프트웨어는 변화를 이끌 핵심기술로 부각되고 있는데(정진주, 2020), 영국, 에스토니아, 핀란드, 인도, 중국, 일본 등의 국가들은 SW 교육의 중요성을 인지하여 초등학교 저학년부터 소프트웨어 교육을 필수화하고 있으며, 전 세계적으로도 소프트웨어 교육을 강화하고 있다(강신천, 안성진, 성영훈, 정영식, 김영애, 서정희, 박세영, 2019).

창의성은 인공지능과 로봇이 많은 부분을 담당하게 되는 미래사회로 갈수록 더욱 중요한 인간의 능력이라고 할 수 있는데, 수학 교육에서도 창의성을 기를 수 있는 과제나 수업에 대해 여러 선행 연구들에서 주목하고 있다(강주영, 김동화, 2020; 문성재, 노정원, 노예솔, 이경화, 2019; Ervynck, 1991; Lee, 2017). 수학적 창의성은 수학 영재가 가지고 있는 대표적인 특성으로, 수학적 창의성에 관하여 황우형 외(2006)는 “새로운 개념을 배우거나 문제를 해결하려고 할 때 기존에 갖고 있는 개념을 연결, 연합하여 새로운 개념을 쉽게 이해하거나 스스로 새로운 개념을 구성하는 능력”으로 정의하였다. 또한 Ervynck(1991)은 수학적 창의성이 논리 연역적인 구조와 개념의 역사적 발달을 고려한 수학적 개념들에 관한 사고의 인지적인 구조를 개발하고 문제를 해결하기 위한 능력이라 하였다.

융합 교육은 다양한 학문 분야에 대한 융합적 능력을 갖춘 인재 양성과 제4차 산업혁명 시대의 영재교육을 위해 필요한데(류성림, 이종학, 윤마병, 김학성, 2018), 수학 영재교육에서도 실세계 현상을 포함한 다양한 맥락에서 학생들이 학습한 수학적 지식들을 다양한 형태로 융·복합적으로 적용하고, 창의적으로 문제를 해결하는 교수·학습에 관심이 집중되고 있다(김유경, 방정숙, 2015). 특히 최인선(2012)은 ‘창의·융합 인재’를 핵심으로 하는 수학교육 정책을 강화해야 하며 수학 영재교육에서 수학적 창의성 신장을 위해서는 분야 간 벽을 허무는 융합 교육이 필요하다고 하였다.

최근 프로그래밍을 활용한 융합 수업이 21세기 핵심 교육으로 주목받기 시작하였다. 이와 관련하여 2015 개정 수학과 교육과정(교육부, 2015a)에서는 4차 산업혁명 시대에 대비하여 정보처리 능력과 공학적 도구의 활용 능력을 강조하고 있으며 제2차 수학교육 종합계획(교육부, 2015b)에서도 체험·탐구 중심의 수학 교육의 일환으로 공학적 도구 활용 지원을 강조하고 있다. 수학과 프로그래밍은 깊은 연관을 맺고 있으며, 수학 영재교육에서 프로그래밍을 활용함으로써 수학 영재의 문제해결 능력을 향상시키고 문제 상황에 대한 긍정적인 시각을 이끌어내는 데 도움을 주며 수학적 능력과 수학적

창의성을 배양할 수 있다(정인우, 조한혁, 2020).

프로그래밍 교육의 핵심 목적은 컴퓨팅 사고(computational thinking)에 있다. 컴퓨팅 사고는 컴퓨터 과학의 근본적인 개념을 사용하여 문제를 해결하고, 시스템을 설계하고, 인간의 행동을 이해하는 것으로서 3R's(Reading, wRiting and aRithmetic)처럼 누구나 갖추어야 할 역량으로 여겨진다(Wing, 2006). 또한 컴퓨팅 사고는 과학, 수학, 예술 등 모든 분야에서 갖추어야 할 기본역량으로써 미래사회의 리더가 될 영재에게는 필수라 할 수 있다(이재호, 장준형, 2020). Wing(2006)은 컴퓨팅 사고의 구성 요소를 크게 추상화(abstraction)와 자동화(automation)로 나누어 설명하였고, 이는 컴퓨팅 사고를 효과적인 문제해결을 위해 문제의 핵심을 추상화하고 이를 자동화하여 표현하는 데 필요한 사고 과정으로 정의할 수 있다.

수학교육에서의 컴퓨팅 사고에 대해 장경운(2017)은 수학 교과의 수학적 문제해결 관련 요소로 보았고, 신동조, 고상숙(2019)은 컴퓨팅 사고와 수학적 사고 간의 관계를 분석하고, 수학적 사고를 기반으로 하는 컴퓨팅 사고의 연구의 필요성을 제기하였지만, 수학교육 연구 분야에서 컴퓨팅 사고의 개념에 관해서는 연구는 아직 초기 단계라 볼 수 있기 때문에 앞으로 추가적인 연구가 더 필요하다.

수학 영재들은 프로그래밍을 활용한 융합 수업을 통하여 수학과 관련된 문제들을 컴퓨팅적 사고로 접근하고 해결하는 과정에서 다양한 문제를 컴퓨팅 사고를 활용하여 바라보고 창의적인 해결책을 모색할 수 있는 훈련이나 경험을 할 수 있고(지현경, 2021). 단순한 기술 수준이 아닌 수학적 창의성과 컴퓨팅 사고와 같은 고등 사고능력을 기를 수 있다. 이러한 필요성에 따라 영재교육에서 컴퓨팅 사고를 향상시키기 위한 소프트웨어 교육, 융합 교육이 활발히 이루어지고 있다(이재호, 장준형, 2020). 하지만, 컴퓨팅 사고와 수학적 창의성을 신장시키기 위해 프로그래밍을 활용한 융합 수업은 미래사회를 대비한 새로운 시도와 장점 및 가능성이 있지만, 수학과 같은 기존 과목과의 유기적인 교과 융합 학습을 수업에 적용하는 것은 쉽지 않은 과제이다(지현경, 2021).

반면 프로그래밍을 활용한 융합 교육에 대한 사례 및 연구는 대부분 컴퓨터 교육학의 맥락에서 정보 및 실과 교과 운영을 위한 소프트웨어 교육 프로그램의 적용 후 효과성을 검증하거나, 교수·학습모형 혹은 수업모형 중심의 연구로 진행되고 있다(지현경, 2021). 프로그래밍을 활용한 융합 수업이 수학 영재들의 수학적 창의성과 컴퓨팅 사고 발현에 어떻게 기여하고 있는지 관찰하거나 분석하는 질적인 연구는 미흡한 실정이며, 수학적 창의성과 컴퓨팅 사고 발현에 효과적인 융합 수업을 위해 전략적인 시사점을 확인할 필요도 있다.

프로그래밍을 활용한 융합 수업에 적용될 수 있는 프로그램 중, Python은 사용자가 만든 프로그램을 바로 실행하여 결과를 확인할 수 있으면서도 문법이 쉽고 간단해서

처음 프로그래밍을 시작하는 사람도 쉽게 배울 수 있는 고급 프로그래밍 언어이다(이도영, 정종인, 2018). 초등학생 소프트웨어 교육에서 엔트리나 스크래치를 주로 사용하고 있는데, 이는 처음 프로그래밍을 접하는 학생들에게 흥미를 유발하기에는 좋지만 실제 고급형 프로그래밍 언어로 넘어갈 때 연계성이 부족하므로 Python은 프로그래밍 언어 간의 징검다리 역할을 할 수 있다(박대륜, 유인환, 2018). 하지만 이러한 장점에도 불구하고 Python 프로그램을 활용한 프로그래밍을 활용한 융합 수업에 관련한 연구는 미흡한 실정이며, 이에 따라 영재교육에서의 활용도 제한적인 상태라 할 수 있다(김예미, 고호경, 허난, 2020).

따라서 본 연구에서는 초등 수학 영재들을 대상으로 Python 프로그래밍을 활용한 융합 수업 프로그램을 개발하고, 실제 수업 적용을 통해 발현되는 수학적 창의성과 컴퓨팅 사고를 분석하고자 하며 다음과 같은 연구 문제를 설정하였다.

첫째, 초등 수학 영재가 Python 프로그래밍을 활용한 융합 수업 프로그램에서 나타내는 수학적 창의성의 특징은 무엇인가?

둘째, 초등 수학 영재가 Python 프로그래밍을 활용한 융합 수업 프로그램에서 나타내는 컴퓨팅 사고의 특징은 무엇인가?

II. 이론적 배경

1. 프로그래밍을 활용한 융합 수업

프로그래밍(programming)은 코드(컴퓨터 언어)를 나열해 프로그램을 만드는 ‘코딩(coding)’의 넓은 의미로 사용된다(김원준, 김은아, 오세경, 2019). 즉, 컴퓨터 언어를 사용하여 프로그램을 만드는 것을 코딩이라 하며, 프로그램을 만들 수 있도록 알고리즘을 교육하는 것이 프로그래밍 교육이다(김원준 외, 2019).

프로그래밍 교육을 통해 코드를 효율적으로 작성하며, 학습자는 내재되어 있는 수학의 원리를 경험할 수 있다. 프로그래밍에서 수학은 융합하기 가장 좋은 과목이고 수학 과에서도 프로그래밍은 수학의 가치를 느낄 수 있는 좋은 소재이므로 프로그래밍과 수학이 적절히 융합된 것이 요구된다고 볼 수 있다(김나리, 서용현, 조한혁, 2018). 이처럼 프로그래밍과 수학은 깊은 연관을 맺고 있으며, 수학교육에서 컴퓨터를 이용하면 기초적인 프로그래밍의 알고리즘을 이해하게 된다(유정호, 2015). 또한 알고리즘과 프로그래밍이 새로운 수학적 대상이 되며, 추상적인 수학적 개념과 실제적인 표상들의 연관성이 지식망을 형성하여 효과적인 학습 환경의 핵심적인 요소가 된다(박성현, 2002; 이승우, 2020).

컴퓨터로 수학을 하는 융합 수업을 할 경우, 다양한 장점들을 기대할 수 있는데, 우선 프로그래밍 학습은 학습자에게 단순한 컴퓨터 언어습득 이상의 것을 제공할 수 있

다. 프로그래밍을 활용함으로써 학습자의 문제 해결 능력을 향상시킬 수 있고, 문제 상황에 대한 긍정적인 시각을 이끌어내는 데 도움을 주며 수학적 능력과 창의성을 배양할 수 있다(이승우, 2020; 황혜진, 2016). 또한 프로그래밍을 통해 수학적 아이디어를 절차적 언어로 표현하며 수학적 문제해결의 과정을 경험할 수 있고, 추상적 수학 개념을 쉽고 구체적으로 접근할 수 있도록 도움을 줄 수 있다(Clement, Battista, & Sarama, 2001). 프로그래밍의 기본이 되는 알고리즘 작성의 논리적 과정을 통하여 학생들이 수학에서 발생하는 다양한 문제를 컴퓨터로 해결하는 직접적인 경험 그 자체가 수학적 문제해결 교육의 실체가 될 것이다(심광섭, 심성아, 2018).

2. 수학적 창의성

수학적 창의성은 수학 분야의 논리 연역적인 특성이 고려된 창의성으로 수학 영재가 가져야 할 매우 중요한 특성으로(강주영, 김동화, 2020), 수학적 창의성에 대한 정의는 크게 두 가지로 구분될 수 있다. 첫 번째는 인지 과정 속에 포함된 사고의 특성에 초점을 맞추어 수학적 창의성을 설명하며 창의적인 사고의 질을 강조하는 경우이며(Haylock, 1987; Krutetskii, 1976), 두 번째는 창의적인 사고의 결과로 나온 산출물에 초점을 맞추어 수학적 창의성을 설명하는 경우이다(권오남, 박정숙, 박지현, 조영미, 2005; Jensen, 1973; Sriraman, 2005). 수학적 창의성은 다양하고 독창적으로 문제를 해결할 수 있는 능력 등을 포함한 수학적 인지 능력이며, 최종 산출물의 관점에서는 새롭고 독창적이며 유용한 산출물을 만들어내는 능력으로 설명될 수 있다.

수학적 창의성의 구성 요소를 언급할 때는, 많은 학자들(김홍원 외, 1997; 남승인 외, 2016; Callahan, 1977; Leikin, 2009; Silver, 1997)은 공통적으로 수학적 창의성의 구성 요소로 유창성, 융통성, 독창성, 정교성, 민감성을 제시했다. 한편 이대현(2012)은 수학적 창의성을 인지적, 정의적, 태도적 요소로 구분하여 제시하기도 하였는데, 인지적 측면은 '수학적 창의성을 계발하기 위한 지적 능력이나 지식, 사고 기술'을 의미하며, 그 요소로는 지식과 경험, 확산적 사고, 논리성, 직관, 메타인지가 있다(김부윤, 이지성, 2007; 이대현, 2012). 두 번째로 수학적 창의성의 정의적 측면은 '수학에 대해 마음 속에 가지고 있는 일관성 있는 신념 체계'를 의미하며, 그 요소로는 흥미, 동기, 지적 호기심, 가치 인식을 들 수 있다(이대현, 2012). 마지막으로 수학적 창의성의 태도적 측면은 '실제로 수학을 하는 과정에 나타나는 자세와 태도'를 의미하며 특히 구성원 간의 토론과 아이디어의 공유, 화합을 통한 수학 학습을 강조하고 있는 교실 환경에서는 학생들이 자신의 생각과 판단을 믿고, 주어진 과제를 끝까지 해결하려는 자세와 태도가 중요하다(이대현, 2012).

수학적 창의성을 계발시키기 위해서 기본적으로 강조되어야 하는 것은 개념의 정확한 학습과 경험이며, 이 속에는 알고리즘, 절차, 패턴 등이 포함되어 있다(이지성,

2006; 이대현, 2012). 이러한 획득을 통하여 통찰과 연결이 일어나며, 더 수준 높은 창의성의 증진을 가져올 가능성이 커진다고 할 수 있는데, 완전한 무의 상태에서 어떠한 새로운 것을 만들어내는 것은 어려우며, 부정확한 기존 개념의 소유는 수학적 활동에서의 창의성의 증진을 방해하기 때문이다(이지성, 2006). 수학적 창의성의 계발 및 육성과 증진에 관한 연구들은 대부분 문제해결의 전략이나 기법을 통하여 수학적 창의성이 길러질 수 있다고 주장하고 있으며, Becker & Shimada(1997)의 open-ended 문제 해결은 수학적 창의성의 육성과 관련이 있다.

3. 컴퓨팅 사고

컴퓨팅 사고(Computational Thinking, CT)는 1942년 Lawrence가 처음 사용한 용어로, 1980년대 이후 PC의 보급이 확대되면서 이를 활용한 교육이 본격적으로 시작되었다(이재호, 장준형, 2018). CT는 Wing(2006)에 의해 본격적으로 정의되었는데, Wing에 따르면 CT는 “컴퓨터과학자처럼 사고하는 것”(Wing, 2006; 2017)으로 정의되며, “컴퓨터 과학의 근본적인 개념을 사용하여 문제를 해결하고, 시스템을 설계하고, 인간의 행동을 이해하는 것”을 의미한다(Wing, 2006). Wing(2006)은 컴퓨팅 사고의 구성요소를 크게 추상화(abstraction)와 자동화(automation)로 나누어 설명하였다. 여기서 추상화는 분해와 명확한 구분 없이 사용되었는데, 문제를 해결 가능한 단위로 만들기 위해 문제에 관련된 속성을 모델링하거나 문제에 적합한 표상을 선택하는 것을 의미한다(신동조 외, 2019).

컴퓨팅 사고와 문제해결은 깊은 관계가 있는데, 국제 교육공학 학회, 컴퓨터 과학교사 협의회(The International Society for Technology in Education [ISTE] and the Computer Science Teachers Association [CSTA], 2011)에서는 컴퓨팅 사고를 ‘컴퓨터 및 기타 도구를 사용하여 문제를 해결할 수 있는 방식으로 문제를 공식화하여 데이터의 논리적 구현 및 분석, 추상화를 통한 데이터 분석을 통해 데이터를 표현하고, 알고리즘적 사고를 통한 해답을 자동화하는 과정을 포함하는 문제해결 과정’으로 정의했다. 컴퓨팅 사고는 사람이 사용하기는 하지만 인지하지 못했던 사고과정에 필요한 다양한 기법들(순차, 반복, 알고리즘 구현 방법 등)을 구체적으로 학습하기 때문에, 문제해결에 보다 체계적으로 접근할 수 있으며 무엇보다도 속도가 빠르기 때문에 문제해결과정에 있어 우위를 점할 수 있다(안성진, 2015). 문제해결을 적용한 프로그래밍 수업을 통해서 컴퓨팅 사고가 향상되었음을 입증하는 연구들도 살펴보면(김영직, 2020; 박주연, 2015; 신수범, 2015; 양재명 외, 2017), 문제해결과 컴퓨팅 사고는 긍정적인 관계가 있는 것을 짐작할 수 있다.

III. 연구 방법

1. 연구 대상

본 연구에서는 경상남도 소재 J초등학교 수학·과학 영재학급 학생 16명(초등학교 4학년 11명, 5학년 5명)을 연구 대상으로 하였다. 연구대상자는 2021학년도에 경상남도 소재 J초등학교 수학·과학 영재학급 학생으로 자기 추천과 관찰 평가를 통해 선정위원회에서 최종 선발된 4~5학년 학생들이다. 16명 전원이 연구 참여자 모집에 신청하여 수업에 참여하였고, 16명의 학생 중 영재교육 경험이 있는 학생은 5명이며 2020년도까지 코딩교육 경험이 있는 학생은 10명, 둘 다 경험한 학생 5명, 둘 다 경험하지 않은 학생 6명이었다. 학생 16명에 대한 학년, 성별, 영재교육 또는 코딩교육 경험 유무에 대한 상세 내용은 [표 1]과 같다.

번호	학년	성별	영재교육 (2020)	코딩교육 (~2020)	번호	학년	성별	영재교육 (2020)	코딩교육 (~2020)
S1	5	남	O	O	S9	4	여	X	O
S2	5	남	O	O	S10	4	여	X	O
S3	5	남	O	O	S11	4	여	X	O
S4	5	여	O	O	S12	4	남	X	X
S5	5	여	O	O	S13	4	여	X	X
S6	4	남	X	O	S14	4	여	X	X
S7	4	여	X	O	S15	4	여	X	X
S8	4	여	X	O	S16	4	여	X	X

* O : 영재교육 또는 코딩교육에 대한 경험 있음; X : 영재교육 또는 코딩교육에 대한 경험 없음

[표 1] 연구대상자의 영재교육 & 코딩교육 경험 유무

2. 연구 절차

본 연구는 사례 연구로, Python 적용 프로그래밍을 활용한 융합 수업에서 나타내는 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고를 알아보려는 질적 연구에 해당한다. 프로그래밍을 활용한 융합 수업 프로그램을 개발하여 적용하는 연구 절차는 [그림 1]과 같다.



[그림 1] 연구 절차

본 연구에서는 연구 문제와 관련된 문헌 연구를 통해 [그림 1]과 같이 Python 적용 프로그래밍을 활용한 9차시 융합 수업 프로그램을 개발하였다. 개발한 Python 적용 융합 수업 프로그램은 교육경력 8년 이상으로 영재교육 관련 석사학위 이상 소지자인 영재교육 현장 전문가 6명의 검토를 받은 후 수정 보완하였다. 2021년 9월 7일과 9월 14일, 9월 28일 3차시씩 세 차례에 걸쳐 연구대상에게 적용하였다. 적용하기 전에 사전 설문을 실시하였고, 적용 후에는 사후 면담을 실시하여 Python 적용 프로그래밍을 활용한 융합 수업 프로그램의 효과를 분석하는 데 활용하였다. 이외에도 학습 과정 중에 나타나는 특성관찰, 활동 결과물, 산출물 등을 바탕으로 학생들의 수학적 창의성과 컴퓨팅 사고를 분석하였다.

3. 자료 수집

본 연구에는 융합 수업 프로그램 활동을 하는 과정에서 나타나는 학생들의 인지적, 정서적, 태도적 특성을 면밀히 관찰하기 위해서 16명의 학생들이 각자 한 대의 태블릿 PC를 가지고 활동할 수 있도록 하였다. 자료 수집을 위해 학생들의 활동을 태블릿 PC의 녹화 기능을 이용하여 태블릿 PC 상에서 이루어지는 Python 학습 과정을 녹화하였다. 그리고 차시 학습 과정이나 문제 해결을 위해 교사와 학생들 간의 전체적인 상호작용 과정과 영재 학생들이 칠판에 제시한 여러 가지 구체적인 표현들도 학생들의 사고 과정을 이해하는데 필요한 자료라고 판단하여 전 차시 수업도 비디오로 녹화하였다. 모든 녹화한 내용은 전사(transcription)하였다.

프로그래밍을 활용한 융합 수업 프로그램을 개발하기 전에 연구 대상인 초등 수학 영재를 대상으로 컴퓨터를 활용한 수학 인식 조사 설문을 서술형으로 실시하였다. 그리고 9차시의 활동이 모두 끝난 후 학생들과 Python 프로그래밍을 활용한 융합 수업에 대한 학생들의 반응을 조사하기 위해 반구조화된 면담을 실시하였다. 연구자는 프로그래밍에 관한 경험과 코딩 방식에 대하여 질문을 던졌으며 수업 녹화 장면에서 연구 대상이 문제를 해결하는 장면에 대하여 구체적인 질문을 던졌다. 면담 내용 역시 모두 녹음하고 전사하였다.

그 외에도 연구의 타당성과 신뢰성을 높이기 위해 면담 자료, 설문, 수업 활동 자료, 학습자의 언어적, 비언어적인 활동, 현장 노트 등 다양한 자료를 수집하였다. 이처럼 여러 방법들을 통합함으로써 각 방법들의 결점을 극복하고 강점을 취할 수 있다.

4. 분석 방법

자료를 분석하는 과정은 다음과 같다. 우선 태블릿과 수업 녹화, 면담 내용을 전사하여 코딩하고, 코딩된 자료를 반복해서 들으며 분석하였고, 분석 결과에 대한 연구자의 해석을 진술할 때에는 이를 뒷받침해 주는 활동 과정과 면담 내용도 근거로 포함시켰다. 또한 수학적 창의성과 컴퓨팅 사고 과정을 분석하기 위해, 녹화 자료 중에서 주목할 만한 특정

구간을 선정하여 30초에서 1분 간격으로 미시적으로 분석함으로써 참여 학생이 Python 언어로 코드를 표현하는 방식, 연구자의 설명을 이해하는 패턴 및 Python 프로그래밍을 하는 활동 패턴에 관한 특성을 [그림 2]와 같이 포착하여 정리하였다.

8:50	9:13	12:37	13:5	14	17:24
S10: 되었으면 좋겠다~ 제발 돼라!	S10: 홀수 변수를 지웠다가 다시 써 봄.	S10: 어떻게 했었지?	S10: ':'을 '='으로 고침 어! 됐다!!	S10: '홀수' 글자도 같이 출력하고 싶어 함.	S10: 음.. 끝! 복사해놔야지. 예~!
흥미, 동기	디버깅	과제집착력	발견, 통찰	동기, 호기심	성취감

[그림 2] 전사 자료를 시간 구간별로 코딩한 예시

융합 수업 프로그램을 실행하면서 나타나는 수학 영재들의 수학적 창의성 발달 단계를 분석하기 위하여, 본 연구에서는 수업 과정과 학생들이 표현한 코드 및 프로그래밍한 결과를 [표 2]와 같이 Ervynck(1991)이 제시한 수학적 창의성의 3수준을 바탕으로 분석하였다. 분석틀의 기본적인 형태는 수정하지 않았지만, 본 연구에서 얻은 학생들의 활동 분류를 명확히 하기 위해 각 수준의 특징을 선행 연구(이경화, 2015; Tall, 1991)를 참고하여 세부적인 내용을 수정 및 보완하여 기술하였다.

수준	요약	설명
1수준	알고리즘 활동	기존 알고리즘을 그대로 적용하여 해결하며, 어떤 알고리즘을 적용하는 것이 좋을지 판단하고, 제대로 수행하는 수준
2수준	창의적(개념·구성적) 활동	알고리즘이 아니라 독자적인 방법으로 추론하여 문제를 해결함. 이미 알고 있는 알고리즘을 활용하는 것보다 문제 상황에서 직접 해결의 단서를 찾고 표현하며 추론하는 수준
3수준	개연적 추측 (직관·통찰) 활동	문제 상황을 직접 통찰하여 해결함. 해결의 단서를 표현할 필요 없이 직관적으로 답을 찾아 나가는 수준

[표 2] 수학적 창의성 수준 분석틀(Ervynck, 1991)

IV. 초등 수학 영재의 융합 수업 프로그램 개발과 적용

1. 초등 수학 영재의 융합 수업 프로그램 수업 모델 설정

초등 수학 영재들에게 수학적 창의성과 컴퓨팅 사고를 계발하며 발현되는 과정을 관찰하기 위하여, 본 프로그램에 프로그래밍을 활용한 융합형 모델과 이야기·동화 모델을 적용하였다.

융합형 모델은 프로그래밍과 수학을 융합함으로써 알고리즘과 프로그래밍이 새로운 수학적 대상이 되며, 추상적인 수학적 개념과 실제적인 표상들의 연관성이 지식망을 형성하여 효과적인 학습 환경의 핵심적인 요소가 될 수 있다(박성현, 2002; 이승우, 2020).

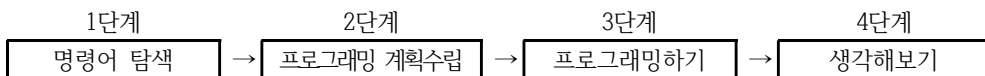
프로그래밍을 수학 영재 수업에 융합함으로써 학습자의 문제 해결 능력을 향상시킬 수 있고, 문제 상황에 대한 긍정적인 시각을 이끌어내는 데 도움을 주며 컴퓨팅 사고와 창의성을 배양할 수 있도록 하였다(이승우, 2020; 황혜진, 2016).

이야기·동화는 학생 스스로 주인공이 되어 문제를 해결할 수 있는 기회를 제공하게 하여 동기 유발과 수학적 태도를 기르는 데 적합하고, 수학의 통합적 사고가 가능하며, 새롭고 가치 있는 창의적인 수학적 산출물을 만들 수 있다(김상룡, 2002; 박재현, 2014). 이야기·동화 모델을 융합 수업 프로그램에 적용하여 스토리텔링에 내재되어 있는 수학적 상황을 발견하고 이를 이용하여 수학을 탐구하게 하였다.

2. 초등 수학 영재의 융합 수업 프로그램 교수 학습 모형 개발

본 연구에서는 Python 프로그램을 적용해 볼 수 있는 교수·학습 모형을 먼저 구성하고 이에 따라서 수학적 창의성과 컴퓨팅 사고가 증진될 수 있는 수업 프로그램을 개발하였다. 수학적 창의성과 관련하여 Ervynk(1991)은 문제해결 과정에서 수학적 창의성이 발현된다고 주장했으며, Silver(1997)는 문제해결 활동을 포함하는 탐구 기반 수학 학습에서 수학적 창의성을 키울 수 있다고 하였고, 수학적 창의성은 ‘문제해결 내에 있는 것’, ‘문제해결과 관련짓지 않고서는 논할 수 없는 것’으로 여겨져 왔다(김판수, 2008; 이경화, 2015). 또한 프로그래밍하는 단계는 일반적으로 문제를 해결하는 과정과 비슷하기 때문에 컴퓨터 프로그래밍은 계산적 사고 기반의 문제해결의 한 유형으로 논의되었다(류희찬, 장인옥, 2010; 이은석, 2013; 전영국, 2015).

본 연구는 Python 언어의 특징과 창의적 문제해결 과정과 컴퓨팅 사고가 발현될 수 있도록 [그림 3]과 같은 1차시 분량의 교수학습모형을 고안했으며, 수업내용에 따라 두 차시에 걸쳐 본 모형이 적용되기도 했다. 그리고 Polya의 문제해결학습모형에 대응되도록 4단계로 구분하였다.



[그림 3] 본 연구의 문제해결학습모형

가) 1단계(명령어 탐색)

전 차시에 학습한 내용과 오늘 학습할 내용과 관련된 개념 및 원리를 이해한 뒤, 간단한 프로그램으로 직접 실행해보며, 필요한 명령어를 탐색하는 단계이다. 이 단계는 수학적 창의력 및 문제해결력을 키우기 위해 프로그래밍 교수의 구성 요소 중 첫째인 프로그램 언어에서 제공하는 특징을 파악하여 필요한 내용을 숙지하는 단계이다(김종진, 2011). Polya의 ‘문제이해’단계(문제의 주요 부분을 주의 깊게 반복하여 여러 측면

에서 살펴보는 단계)에 대응된다.

나) 2단계(프로그래밍 계획수립)

문제에 대한 이해, 이전에 얻은 지식, 과거의 경험으로부터 계획을 세우고, 과정 단계별로 손 코딩하는 단계이다. Polya의 '계획수립'단계(계획을 세우는 단계)에 대응된다.

다) 3단계(프로그래밍하기)

손 코딩한 내용을 기억하며, 각 단계를 정확하게 수행하고 디버깅하는 단계로 인내심이 요구된다. Polya의 '계획실행'단계(문제해결을 위한 계획을 스스로 세우며, 수립한 계획을 잊어버리지 않고 각 단계를 정확하게 수행하는 단계)에 대응된다.

라) 4단계(생각해보기)

완성된 코딩 내용을 발표하고, 잘되거나 아쉬운 점, 친구들의 코딩 결과를 공유함으로써, 획득한 지식을 견고하게 하고 코딩의 간결화와 효율화를 추구할 수 있다(서용현, 2019). 문제를 해결하는 능력을 발달시킬 수 있는 내용을 포함하며, Polya의 '반성' 단계(완성된 문제해결 방법을 검토하고, 획득한 지식을 견고하게 하는 단계)에 대응된다.

3. 초등 수학 영재의 융합 수업 프로그램의 실제

본 연구의 목적은 프로그래밍을 활용한 융합 수업 프로그램이 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고에 어떤 교육적 효과를 지니는지 알아보려고 하는 데 있다. Python은 명령문을 입력하여 프로그램을 작성해야 하므로, 명령어의 단계적 교육을 통해 결과물을 더욱 발전시켜 나갈 수 있도록 수업을 설계하였다. Python의 기초 문법 및 명령어들을 예제들을 통해 익히고, 실제적인 문제들(게임 만들기, 구구단 만들기, 최대공약수 구하기 등)을 해결하도록 함으로써 학습자의 참여도와 성취도를 높이고자 하였다.

해당 학생들은 Python 프로그래밍을 활용한 융합 수업 전에 이미 Python의 의미와 사용법, 기본 명령어 및 출력 방법을 탐구해본 바 있다. 따라서 명령어에 대해 자세히 설명하는 대신, 각 차시 도입 부분에 명령어 퀴즈, 스토리텔링으로 명령어를 간단하게 복습하는 내용으로 구성하였다. 스토리텔링은 '15소년 표류기'를 소재로 하여 '16소년 표류기'를 제목으로 하여 그 안에 문제 상황이 내포되도록 구성하였다.

교육 주제는 프로그래밍의 난이도와 초등 영재 학생들의 수준을 고려하여 변수의 활용, 반복문, 조건문, 함수의 순서로 구성하였으며 총 9차시의 수업 프로그램을 개발하고 설계하였으며, 선정된 주제와 차시별 구체적인 내용은 [표 3]과 같다.

차시	주제	내용
1차시	변수 사용하기	· 변수의 개념 이해하고 변수에 값을 저장하는 기호 알기 · 나만의 변수와 산술연산자를 이용하여 프로그램 만들기
2차시	명령어 익히기	· 비교연산자, 조건문, 반복문 등 명령어 익히기 · 리스트 자료구조의 의미와 사용법 알고 짝수, 홀수 분류하기
3차시	반복문 사용하기	· 제어문(반복문)을 사용하여 구구단 만들기
4차시	조건문 사용하기	· 제어문(조건문)을 사용하여 숫자를 구별하는 프로그램 만들기
5차시	함수 이해하기	· 검은 상자에 숫자가 들어가면 어떤 관계로 숫자가 나오는지 파악하며 함수의 역할 이해하기 · 임의의 수를 뽑는 랜덤 함수를 활용한 주사위 시뮬레이션
6차시	게임 만들기	· 친구와 할 수 있는 재미있는 게임이나 퀴즈 만들기
7차시	함수 정의하고 호출하기	· def 명령어를 활용하여 함수를 정의하는 방법 알기 · 수학적 원리가 포함된 함수를 만들고 def 명령어를 사용하여 표현하기
8차시	최대공약수 구하는 프로그램 만들기	· 최대공약수 개념 이해 및 유클리드 호제법 알기 · 최대공약수 구하는 방법 탐색하기
9차시		· 최대공약수 구하는 프로그램 만들기

[표 3] 본 연구의 융합 수업 프로그램 개요

V. 초등 수학 영재의 융합 수업 프로그램 효과 분석

1. 수학적 창의성

본 연구의 프로그래밍을 활용한 융합 수업 과정과 초등 수학 영재들이 표현한 코드 및 프로그래밍한 활동 결과를 바탕으로 수학 영재들의 수학적 창의성을 분석하였다. Ervynck (1991)이 제시한 수학적 창의성의 3수준과 수학적 창의성의 구성 요소인 인지적 요소, 정의적 요소, 태도적 요소를 기준으로 분석하였고, 문제해결과정과 산출 결과에서 기존 알고리즘을 그대로 적용하여 해결하는 1수준을 제외한 2수준 이상의 수학적 창의성을 관찰할 수 있었던 학생들을 중심으로 기술하였다.

가. 수학적 창의성 수준 분석

수학적 창의성의 수준 분석에 앞서, 융합 수업 프로그램의 목표가 초등 수학 영재들의 수학적 창의성의 계발이고, 교사의 주도하에 제시된 프로그래밍을 제외하면 그 속에는 학생들의 수학적 창의성과 그 탐구 과정이 포함되어 있기 때문에 연구자는 이것들을 모두 관찰하는 것이 의미 있다고 판단했다. 따라서 교사의 도움 여부와 관계없이 학생들의 프로그래밍 결과 전부를 창의성 수준 분석에 포함시켰으며, 해결한 문제마다 연구자의 도움이 어떤 단계에서 어느 정도 필요했는지 ‘교사 도움 없음’, ‘교사 도움 약간’, ‘교사 도움 중간’, ‘교사 도움 결정적’와 같이 나누어 기술하였다(박재현, 2014).

이상 논의한 내용을 바탕으로 융합 수업 프로그램에서 나타난 초등 수학 영재들의 수학적 창의성 수준과 교사 도움 정도를 [표 4]와 같이 빈도로 기술하였다. 이때 빈도의 합을 연구대상자 수인 16으로 통일하기 위해, 협력하여 문제해결을 한 학생들의 결과도 각각의 결과로 따로 계산하여 기술하였다.

차시	수학적 창의성 수준 빈도			교사 도움 빈도			
	수준 1	수준 2	수준 3	없음	약간	중간	결정적
1차시	15	1	0	15	0	1	0
2차시	14	2	0	12	2	2	0
3차시	16	0	0	12	1	3	0
4차시	13	3	0	8	2	3	3
5~6차시	10	3	3	5	2	6	3
7차시	16	0	0	10	3	3	0
8~9차시	13	2	1	2	4	7	3
계	97	11	4	64	14	25	9

[표 4] 초등 수학 영재들이 융합 수업 프로그램에서 나타낸 수학적 창의성 수준

학생들의 수학적 창의성 수준은 제시되는 문제의 내용과 형태에 따라 그 정도가 다르게 나타났다. 5~6차시의 수업에서 학생들이 흥미 있어 하는 게임을 개방형 문제로 제시하였는데, 학생들이 자신만의 게임을 프로그래밍하면서 비교적 높은 수준의 수학적 창의성이 관찰되었다. 반면에 구구단을 프로그래밍하는 3차시와 나만의 함수를 프로그래밍하는 7차시는 프로그래밍 결과 대부분이 수학적 창의성의 1수준이었다. 문제 형태는 각각 폐쇄형, 개방형으로 제시되었지만, 학생들이 기존에 학습했던 명령어와 알고리즘을 바탕으로 해당 문제를 쉽게 해결할 수 있었기 때문으로 판단된다. 그래서 추후 분석에는 수준 1에 해당하는 프로그래밍 결과는 제시하지 않았다. 여기서 생략한 수준 1의 풀이는 교사의 도움 없이 대부분의 학생들이 독자적으로 코드를 표현할 수 있다.

수학적 창의성 발현의 관점에서는 교사의 도움이 중간 이하이면서 수준 2와 수준 3에 해당하는 문제해결이 중요한데(박재현, 2014), 이에 해당하는 학습 프로그래밍 결과를 차시별로 정리하면, [표 5]와 같다.

3, 7차시는 교사의 도움이 중간 이하이면서 수준 2, 수준 3에 해당하는 프로그래밍 결과가 도출되지 않았는데, 수업의 구성 내용이나 제시된 문제의 형태에 따라 수학적 창의성 수준 차이가 나타나는 것을 관찰할 수 있다. 반면 수준 3의 프로그래밍 결과가 도출된 경우 교사의 도움 정도가 약간 이상인데, 교사의 조언이 학생의 과제 해결과 높은 수준의 수학적 창의성 발현에 어느 정도 도움이 된 것으로 판단된다.

차시	학생 프로그래밍 요약	창의성 수준	교사 도움 정도
1차시	· 변수를 하나 더 추가하여 두 개의 변수로 물건의 개수 당 가격을 출력하도록 함	수준 2	교사 도움 중간
2차시	· 리스트 자료구조를 활용하지 않고, 짝수와 홀수가 세로로 출력되도록 함 · append 함수를 활용하여, 짝수와 홀수가 리스트에 추가되어 출력되도록 함	수준 2	교사 도움 없음
3차시	없음		
4차시	· 6의 약수는 6에 주어진 숫자를 나누었을 때 나머지가 0이라는 특성을 바탕으로, 주어진 숫자가 6의 약수인지 판단하도록 함 · 7의 배수는 주어진 숫자를 7로 나누었을 때 나머지가 0이라는 특성을 바탕으로, 주어진 숫자가 7의 배수인지 판단하도록 함	수준 2	교사 도움 중간
5~6차시	· 로또 숫자 6개를 랜덤으로 출력하게 하는 아이디어를 떠올리며, 변수, 랜덤함수, 리스트, 반복문, append 등 필요한 명령어를 직관적으로 조합하여 완성함	수준 3	교사 도움 약간
	· 숫자 1에서 5까지의 범위에서 랜덤으로 숫자 1개를 출력하게 하여 당첨시키는 아이디어를 떠올리며, 변수, 랜덤함수, 리스트, 반복문, append 등 필요한 명령어를 직관적으로 조합하여 완성함	수준 3	교사 도움 중간
	· 랜덤으로 출력된 숫자가 짝수이면 당첨되는 프로그램을 변수와 조건문, 랜덤함수를 활용하여 완성함.	수준 2	교사 도움 중간
	· 챗봇 아이디어를 떠올려, 조건문을 활용하여 방탈출 게임을 완성함	수준 2	교사 도움 없음
7차시	없음		
8~9차시	· 유클리드 호제법으로 최대공약수를 구하기 위해 큰수에서 작은수를 나누어 나머지가 0임을 판별하는 과정에서 '큰수=작은수, 작은수=나머지'를 직관적으로 파악하여 프로그래밍을 완성함 · 주어진 두 수의 약수를 모두 구한 후, 최대공약수를 찾는 방법으로 프로그래밍함	수준 3	교사 도움 중간
		수준 2	교사 도움 약간

[표 5] 교사 도움 중간 이하이면서 수준 2, 3에 해당하는 학생 프로그래밍 결과

수학적 창의성 발현의 관점에서 수준 2와 수준 3에 해당하는 학생의 프로그래밍 결과와 이에 대한 분석 자료 중 일부는 [표 6]과 같다.

<5~6차시> random 모듈, randint 함수, 산술연산자(더하기, 빼기, 나누기, 곱셈) 등을 사용하여 친구와 할 수 있는 재미있는 게임을 만들어 봅시다.

<p>S2(좌), S9과 S10(우) /수준 3 /교사 도움 약간(좌) 교사 도움 중간(우)</p>	<p>[수준] S2, S9와 S10은 다른 프로그래밍의 지식 없이 기존에 알던 명령어를 직관적으로 조합하여 생각해낸 아이디어를 해결해 나갔다. 리스트에 숫자를 추가하여 출력하기 위해 반복문과 append와 random.randint 명령어를 활용하며 문제 상황에서 필요한 코드를 바로 추론한 것이라 할 수 있다. 이러한 과정은 단순히 알고리즘을 나열하거나 조합한 수준을 넘어서 통찰을 통해 코드를 표현하는 수준이므로 수준 3에 해당 한다고 할 수 있다.</p> <p>[교사 도움] S2는 자신의 아이디어를 교사에게 설명하였고, 교사는 참신한 아이디어임을 암시하는 반응을 제공하였으며, S2는 표현한 코드를 약 15분 동안 디버깅 과정을 거쳐 프로그램을 완성하였다. S9와 S10은 서로 협력하여 프로그래밍을 완성하였고, 최초로 리스트에 랜덤으로 출력되는 숫자를 추가하는 시도를 한 것은 S10였다. 이후 S10이 S9에게 같이 탐구해 볼 것을 권하여 협력하였고, 학생들은 리스트의 위치에 대해 헛갈려 하자, 교사는 '당첨 = []'의 위치가 반복문 앞에 제시되도록 도움을 주었다.</p>
<p><8~9차시> 190과 722의 최대공약수를 여러 가지 방법으로 구해봅시다.</p>	

<p>S2(좌), S4와 S5(우) /수준 3(좌), 수준 2(우) /교사 도움 중간(좌) 교사 도움 약간(우)</p>	<p>[수준] S2는 유클리드 호제법으로 최대공약수를 구하는 코드를 표현하였는데, 이러한 알고리즘을 미리 알고 있지 않았기 때문에 수준 1로 보기 어려우며, 큰 수에서 작은 수를 나누는 식이 나머지가 0이 될 때까지 반복하는 과정에서 큰 수는 작은 수가 되고, 작은 수는 나머지가 된다는 사실을 직관적으로 알고 표현하고자 하였으므로 수준 2로 보기에 어려움이 있다. S4와 S5는 190과 722의 약수를 각각 구하기 위해, 약수를 구별하는 알고리즘을 나열하여 제시하여 이들의 최대공약수를 구하는 2수준의 추론 과정을 보였다.</p> <p>[교사 도움] S2는 큰 수 나누기 작은 수가 반복되는 것이 멈춰야 한다는 사실을 인지하지 못하여 교사가 힌트를 주어 'break'를 생각해냈으며, '큰수= 작은수, 작은수=나머지'를 교사가 제시하여 프로그램을 완성하였다. S4와 S5는 자신들이 쉽게 할 수 있는 약수를 구하는 프로그램을 활용하겠다는 아이디어에 교사는 긍정적인 반응을 보였고, 서로 협력하여 프로그램을 완성하였다.</p>
--	---

[표 6] 학생 프로그래밍 분석 결과

초등 수학 영재들의 프로그래밍 결과에서 수학적 창의성 발현의 요인으로는, 첫째 학생들의 수학적 창의성 발현에 자극을 준 개방형 문제를 꼽을 수 있다. 사실 기본적인 명령어를 학습하면서 프로그래밍을 경험하는 본 수업의 특성상, 연구자는 매 차시에 개방형 문제를 제시할 수 없었다. 하지만 해결 방법이 열려있는 개방형 문제에서

학생들은 각자의 수준에서 새로운 프로그래밍 결과를 도출하는 모습을 보였다.

두 번째로 학생들의 수준에 적합하고 흥미로운 문제가 학생들의 수학적 창의성 발현의 원동력이 되었을 것으로 판단된다. 학생들의 수준에 적당히 도전적이면서 게임의 요소가 추가된 문제를 제시하였을 때 수학 영재들에게서 다소 높은 수준의 수학적 창의성을 관찰할 수 있었다.

세 번째로 수학적 의사소통이 수학적 창의성 발현에 도움이 되었을 것으로 판단되는데, 두 학생이 협력해서 의논하는 과정에서 약수와 배수의 개념을 인식하고 구성하며, 이를 코드로 표현하기 위해 아이디어를 공유하고 반성하는 등의 모습을 관찰할 수 있었고, 프로그래밍 결과에서도 높은 수학적 창의성 수준이 발현되는 것을 관찰할 수 있었다.

마지막으로 수학적 개념과 알고리즘, 절차 등에 대한 정확한 학습과 경험이 수학적 창의성 발현에 도움이 되었을 것으로 판단된다. 학습한 알고리즘과 수학적 개념이 다양하고 정확할수록 코드를 표현하여 문제를 해결하는 수학적 창의성의 수준이 높은 것을 알 수 있었는데, 지식의 정확한 획득을 통하여 통찰과 연결이 일어나며, 더 높은 수준의 수학적 창의성이 발현될 것으로 사료된다.

본 연구 결과를 바탕으로, 수학적 창의성이 수준 2 이상 발현된 학생들을 중심으로 수학적 창의성에 있어서 본 수업의 의미를 보다 심층적으로 알기 위해, 수학적 창의성 요소인 인지적 요소, 정의적 요소, 태도적 요소로 나누어 분석하고자 한다.

나. 인지적 요소

프로그래밍도 기본적으로 아이디어가 있어야 가능한데(전영국, 2015), 학생들은 기존에 알고 있던 수학적 지식과 수학 문제를 다루는 방식을 프로그래밍 작업에 자연스럽게 적용하였다. S3은 문제를 읽어보고 이해하면서 배수를 구별하는 프로그램을 만들기 위해 배수의 성질을 떠올려 손코딩을 해나갔으며, 2의 배수를 구하기 위해 주어진 수에 2를 나누었을 때 나머지가 0이 되면 그 수는 '2의 배수'라는 사실을 [표 7]과 같이 스스로 인지하여 프로그래밍을 완성하였다.

S3 : 선생님, 2의 배수 하고 싶은데, 변수에 2를 나눴을 때 계속 나누어져야 할 것 같은데 어떻게 해야 해요?

연구자 : 변수에 2를 나누면 나머지가 어떻게 되어야 하지?

S3 : 아! 알겠어요~ 나머지가 0이요.

(나머지를 구하는 명령어 자료 찾아보며 혼잣말로) % 네.

[표 7] 이미 알고 있는 지식을 활용·재구성한 사례 1

한편 5~6차시 게임 만들기 수업에서 S12는 1에서 35까지의 숫자 중 무작위로 도출

된 수가 짝수이면 당첨, 홀수이면 낙첨시키는 변형된 로또 게임을 만들었다. [표 8]은 수업 후 면담에서 자신이 만든 게임을 설명하는 모습으로, 본 수업에서 배웠던 명령어들을 재구성하고 기존에 알고 있던 짝수와 홀수의 개념을 적용하여 [표 8]과 같이 간단하면서도 독창적인 게임 프로그램을 만들었다.

S12 : 숫자가 나타나고, 짝수이면 당첨, 홀수이면 낙첨으로 나타내게 하고 싶었어요. 여기서 숫자는 랜덤으로 나타내게 해요. 1에서 35까지 중에.. 저번에 배웠던 if, else로 만들었어요.
 연구자 : 짝수, 홀수는 어떻게 구별했나요?
 S12 : 2가 나누어지면 짝수, 안 나누어지면 홀수로 했어요.

[표 8] 이미 알고 있는 지식을 활용·재구성한 사례 2

4차시 숫자 구별하는 프로그램 만들기 수업에서 S6은 3가지 프로그래밍을 산출하며 유연성을 보였는데, 특히 약수를 구별하는 프로그램을 코딩하는 과정에서 통찰력을 엿볼 수 있었다.

수업 전사 내용	코딩 내용
S6 : 약수 한번 해 볼까? 음.. 어떻게 해야 하지? 우선 출력을 바꾸고!	
S6 : 곱하기로 해야 하나? 아닌데. 나누기로 해서 나머지로 해야 할 것 같은데 나누기로 해 볼까? 나눠봐야겠어. (2분 후) 음.. 숫자랑 6이랑 위치를 바꿔보자! 됐다 됐다! 다 됐다 예~	

[표 9] 직관과 통찰을 바탕으로 한 새로운 문제해결 사례

[표 9]와 같이 S6은 먼저 코딩한 배수를 구별하는 프로그램을 골똘히 보다가 갑자기 약수를 구별하는 아이디어가 떠올라 쉽게 문제를 해결하는 모습을 보였다. 표현한 코드의 숫자 위치를 바꿔보는 디버깅 과정에서 알고자 하는 숫자를 6으로 나누었을 때 나머지가 0이 되면 그 숫자는 6의 약수라는 사실을 갑자기 떠올린 것이다.

[표 10]과 같이 S4와 S5는 손 코딩과 디버깅 과정에서 잘되지 않을 경우에 다시 처음 부분으로 되돌아가서 천천히 살펴보면서 다른 방법으로 프로그래밍을 하다 오류를 발견하고 문제를 해결하는 모습을 보였다.

S5 : 챗봇 만들어볼까?
 S4 : 재밌을 것 같아. 한번 해 볼래.
 S5 : (생각 중) 뭐라고 만들어야 될까?
 S4 : 막혔다..막혔다..
 S5 : 랜덤함수 쓰고.. 아니면 우리 아이템을 랜덤으로 뽑을 수 있게 하는 거 어때?
 일단 아이템을 먼저 여기에다 적어야겠다.
 ...
 S5 : 이게 변수로 가는 거 아닌가?
 S4 : 그게 왜 변수로 가?
 S5 : 안 되겠다. 처음부터 다시 해 보자.
 ...
 S5 : 지금 그걸 넣을 이유가 없어. 아 지금 두 번째 줄이 잘못되었네.
 오 됐다! 됐다!

[표 10] 반성적 사고 능력 사례

다. 정의적 요소

수업에 참여하는 데에 대한 재미와 즐거움, 내적 동기는 창의성의 중요한 요소이며, 특히 수학에서 창의적이기 위해서는 수학에 흥미를 가지고 몰입하는 것이 필요하다(이대현, 2012).

S2는 쉽게 명령어의 역할을 이해하고 정확하고 빠르게 문제를 해결하며, 기존에 알고 있던 명령어들을 융합하여 창의적인 코딩 결과를 도출한 학생이었다. [표 11]의 면담 내용을 보더라도 S2는 Python을 활용한 본 수업에 흥미를 느끼고 있으며, 남과 다르게 프로그래밍하고 싶은 내적 동기도 갖고 있음을 알 수 있다. 실제로 2차시의 짝수 홀수 구별하기 문제는 리스트로 간단히 프로그래밍할 수 있지만, S2는 [표 11]에서 같이 append를 추가로 사용하여 남들과는 다르게 짝수와 홀수를 구별하며 문제를 해결하는 모습을 보였다.

짝수 홀수 구별하기 관련 면담 내용	짝수 홀수 구별하기 코딩 내용
S2 : 처음에는 리스트로만 짝수 홀수를 구별했는데, 조금 다르게 하고 싶어서 append 추가해서 넣어봤어요. ...	
S2 : 예전에 코딩 수업해봐서 잘할 수 있었던 것 같아요. 근데 엔트리랑 달라서 더 재밌었어요.	

[표 11] 수업에 대한 흥미와 동기 사례

지적 호기심은 주변의 사물이나 현상에 대해 의문을 갖고 끊임없이 질문을 제기하는 성향을 의미하며, 수학적으로 창의적이기 위해서 지적 호기심은 만족스러운 결과를 얻을 수 있는 내적 동기 유발의 바탕이 될 수 있다(김연주, 1997; 문용린, 2010).

S6은 [표 12]와 같이 수업 중간에 연구자에게 궁금한 것에 대해 질문을 꾸준히 하며 수업 내용에 대한 호기심을 보였는데, 당시 숫자 구별하기 프로그램을 다양하게 제시하며 유창성을 보이기도 했다.

S6 : 약수를 구별하고 싶은데, 12의 약수는 1, 2, 3, 4, 6, 12 이렇게 구하는 거 맞죠?

...

S6 : 선생님~ 약수 다했는데, 배수해 봐도 돼요?

...

S6 : 선생님! 배수 성공했어요! 변수랑 숫자 바꾸니 되었어요!

짝수 홀수도 해보고 싶은데, 해볼게요!

[표 12] 수업에 대한 지적 호기심과 가치 인식 사례

라. 태도적 요소

수학적 창의성의 태도적 영역으로 학생들이 실제로 문제를 해결하는 과정에서 과제에 대한 몰입과 타인의 간섭이나 평가에 구속되지 않는 독립심, 새로운 문제에 도전하려는 도전 의식과 그에 따른 성취감 등을 관찰할 수 있었다.

S4와 S5는 5~6차시 수업에서 독창적인 게임 프로그래밍을 보였는데, 시간 부족으로 게임 프로그래밍을 마무리하지 못하였으나 남아서 끝까지 완성하는 모습을 볼 수 있었다. 이들에게서 디버깅 작업에서 집중력과 과제를 완성하고자 하는 끈기와 제대로 실행될 때까지 작업에 집중하는 과제 집착력을 관찰할 수 있었다.

디버깅 과정에서 직관과 통찰을 보여준 S12은 사전에 프로그래밍 경험이 없었던 학생이었다. 하지만 줄곧 혼자서 집중력을 유지하면서 코딩 작업을 전개하였는데, 에러가 발생하자 연구자가 제시한 명령어 자료를 참고하면서 스스로 해결해 나가고자 하였다. S12는 수업 후 면담 과정에서 [표 13]과 같이 이야기하였는데, 평소 어려운 문제를 해결하는 인내력이 있음을 엿볼 수 있었다.

S12 : 저는 게임 프로그래밍할 때 재밌었지만, 좀 어려웠던 것도 있었어요. 랜덤으로 뽑힌 숫자를 출력하게 하는 것이 헛갈려서 선생님이 주신 자료도 찾아보고, 선생님께도 여쭙보고 했는데 해결되어서 좋았어요. ...
저는 평소에 문제가 안 풀리면, 막 여러 가지 해 보면서 풀려고 노력하는 편이에요.

[표 13] 태도적 요소 사례 1

또한 수업 장면 분석 결과 S12는 수업 초반에 종이에 혼자 손코딩을 차근차근 약 10분에 걸쳐서 하였고, 옆자리 친구와 상호작용을 별로 하지 않고서 혼자 힘으로 문제를 해결하였다. 그가 작성한 소감문에서 [표 14]와 같이 자신의 판단과 생각을 믿고 자신만의 아이디어를 생각해내기 위해 노력하고, 새로운 문제에 도전하고자 하는 도전

의식과 이를 이룬 후 성취감을 느끼고 있음을 알 수 있었다.

S12 : 나는 내가 만든 게임을 친구들이 재밌어하니, 뿌듯했다. 시간만 더 있었으면 많은 명령어를 사용해서 더 재미있는 프로그램을 만들고 싶었다. 또 최대공약수를 구하는 방법에 대한 설명을 들으면서 내가 조금 더 다르게 바꿔보고 싶었다. 하지만 시간이 부족했고, 다음에 또 이 수업을 하게 된다면, 내가 만든 방법도 해 보고 싶다. Python을 이용해서 많은 것을 한다는 생각이 들었고, 코딩을 향한 내 마음이 더 깊어져 계속 배우고 싶다는 생각을 하였다.

[표 14] 태도적 요소 사례 2

이러한 과제집착력, 독립심, 도전감 등이 외적인 간섭과 평가에서 벗어나 자신만의 아이디어를 산출하는 데 영향을 주는데(이대현, 2012), 본 연구에서도 나름의 창의적인 산출물을 제시한 학생에게서 주어진 문제에 흥미와 동기를 가지고 끝까지 해결하려는 도전적인 자세와 태도를 관찰할 수 있었다.

2. 컴퓨팅 사고

컴퓨팅 사고(CT)는 창의적 사고와 같은 4차 산업혁명 시대가 요구하는 역량과 맞물려 있으며, 모든 과학이 수학에 기초하듯 프로그래밍 역시 수학에 기초하기 때문에 CT는 본연적으로 수학적 사고에 의존한다고 할 수 있다(Wing, 2006). 본 연구에서는 수학과 프로그래밍, 수학적 창의성, CT 간의 유의미한 관계를 인식하여, 추상화와 자동화(Wing, 2006)를 중심으로 학생들의 문제 해결 과정 속 CT, 알고리즘 사고, 절차적 사고, 반복적 사고, 비판적 사고, 논리적 사고를 함께 관찰하고 분석하여 기술할 것이다.

가. 추상화

본 연구에서는 CT의 관찰 및 분석을 위해 각 차시마다 자신의 활동을 필기구를 이용하여 알고리즘으로 표현하며 손코딩 한 후 이를 Python에 표현하는 활동을 하게 하였다. 그리고 [그림 4]와 같이 스토리텔링 문제를 절차적으로 추상화하기를 함으로써 불필요한 요소를 제거하고 수업의 핵심 요소를 추출하는 활동을 하게 했더니, 스토리텔링 속 핵심 요소를 ‘변수=식량을 보관하는 기간’, ‘식량이 상하는 기간은 7일 후’로 인지하여 전체 과정을 순서대로 정리하는 모습을 관찰할 수 있었다. 우선 식량을 보관하는 기간을 입력받아 컴퓨터 공간에 저장하고, 제어문(반복문, if-else-elif)을 활용하여 입력된 값이 7일을 초과하는지 판단해서 초과하면 ‘이미 상했어요.’라고 출력하고, 7일과 같으면 ‘지금 바로 먹어야 돼요.’, 그렇지 않다면 ‘먹어도 안전해요.’라고 출력함으로써 문제해결의 순서를 생각하고 적용

하는 절차적 사고(Procedural thinking) 모습과 추상화 과정을 함께 볼 수 있었다.

<스토리텔링 내용>

소년들은 열매를 따거나 사냥을 해서 식량을 얻었어. 날씨가 따뜻해져서 식량이 빨리 상할 것 같아. 배탈이 나지 않도록 식량을 구분해 보자.

[그림 4] 스토리텔링 문제를 절차적으로 추상화하기

이후 문제해결과정에서도 모델링 과정을 관찰할 수 있었는데, [표 15]는 4차시 숫자 구별하기 프로그램과 7차시 나만의 함수 만들기 수업 결과이다. 이들은 입력, 출력, 함수, print, for 등의 프로그래밍 관련 전문적 용어를 사용하며 절차적 사고를 거쳐 자신의 추상화 과정을 설명하였는데, S1은 정수 a가 정수 b로 나누어질 때, a는 b의 배수라는 개념을 정확히 알고, 배수를 구별하기 위한 핵심 요소인 ‘÷’와 ‘나머지=0’을 찾고 필요한 과정만 표현하며 순서대로 정리하고 손 코딩하였고, ‘print’ 철자가 잘못되어 디버깅한 사실을 연구자에게 설명하였다. S7은 수업 후 면담에서 더하기 함수를 만들기 위해 자신이 프로그램을 추상화한 과정을 구체적이고 정확하게 연구자에게 설명하였다. S7은 더하기 함수를 만들면서 1부터 10까지 더하기 계산이 반복적으로 이루어지는 패턴을 인식하고 반복문을 활용하는 반복적 사고를 하는 모습이 관찰되었다.

수업 전사 내용	학생 활동 내용
S1 : 저는 7의 배수를 구별하는 프로그램을 만들었어요. 우선 배수를 구할 수를 입력하고, 그 숫자가 7로 나누었을 때 나머지가 0이면 “7의 배수”, 그렇지 않다면 “7의 배수 아니야”라고 출력하게 했어요. ... 손 코딩에서는 제일 밑에 print 철자가 잘못되어서 실제로는 고쳐서 실행시켰어요.	
S7 : 저는 더하기 함수를 만들었어요. 숫자를 넣으면 그 숫자에 1부터 10까지 더해져서 출력되도록 했어요. 이때 계속 더하기 때문에, for로 하는 반복문을 사용했어요. 함수에 숫자 넣어서 출력해 보니, 1부터 10까지 더해져서 나왔어요.	

[표 15] 문제해결과정에서의 학생들의 추상화 예시

본 연구에서는 학생들에게 문제를 해결하기 위해 수행되는 단계를 순서도로 나타내는 알고리즘 과정을 경험하게 하였다. 특히 마지막 최대공약수를 구하는 수업에서는 알고리즘을 구성하는 방법이 여러 가지가 있을 수 있음을 학생들에게 알려주었고, [그림 5]와 같이 최대공약수를 구하는 다양한 방법을 단순화하여 알고리즘으로 구성하게 하여 비교 후 더 편리한 방법을 선택하게 하였다. 이때, 학생들은 표현된 코드나 시간이 짧게 걸리는 방법을 더 편리하고 효율적인 방법이라고 생각하였다. 최대공약수를 구하기 위해서 프로그램의 정확성과 효율성만을 평가하는 것이 아니라 간결함과 단순성을 고려하는 모습도 볼 수 있었다.

[그림 5] 최대공약수 구하는 여러 가지 방법을 논리적으로 단순화하기

Futschek(2006)은 알고리즘 사고와 창의성이 매우 강한 상관관계를 갖는 능력이라고 정의하였고, 알고리즘 사고는 문제를 해결하기 위해 특정한 해결 방법을 습득하는 사고 양식이라고 하였고, 알고리즘 사고는 수학적 사고와 컴퓨팅 사고 사이에서 연결고리의 역할을 하는 것으로 판단되곤 한다(김병수, 2014 재인용; 이승우, 2020).

[그림 5]와 같이 간단한 최대공약수를 구하는 과제에 대해서도 유클리드 호제법으로 최대공약수를 구하는 코드를 표현하거나 190과 722의 약수를 각각 구하기 위해, 약수를 구별하는 알고리즘을 나열하여 제시하여 이들의 최대공약수를 구하는 등의 다양한 표현 방법이 존재하고, 학습자가 이 중 하나의 방법을 선택하여 정확한 코드로 표현하는 일종의 알고리즘 사고(Algorithmic thinking) 과정에서 각각 수준 3과 수준 2의 수학적 창의성을 관찰할 수 있었다.

알고리즘 사고와 관련하여 유중현과 김종혜(2008)는 문제를 해결하기 위해서는 직관적 사고가 필요하며, 직관적 사고에 의해 나온 해결책들을 통해 결과를 예측

해 보는 추론적·논리적 사고가 필요하며 각각의 해결책들의 장단점을 파악하는 비판적 사고가 이미 포함되어 있다고 말하고 있다. [표 16]과 같이 최대공약수를 구하는 다양한 알고리즘을 탐색해보게 하는 과정에서 S4는 최대공약수를 구하기 위한 3가지 방법 중 방법 1의 불편한 점과 방법 3의 효율성에 대해 연구자에게 정확히 설명하며 비판적 사고(Critical thinking)가 관찰되곤 하였다.

S4 : 방법 3을 선택한 이유는 코드가 짧아서 시간이 얼마 걸리지 않아 편리하다고 생각했어요.
 방법 1은 프로그램 돌린 후에 또 최대공약수를 찾아야 하니 불편하구요..

[표 16] 면담과정에서의 학생들의 비판적 사고 예시

종합하면 학생들은 프로그래밍 계획 수립 단계에서 수학 문제를 최대한 작은 내용 단위로 분해하여 정리하는 경험을 가질 수 있었다. 예를 들어, ‘7의 배수를 구하기 위해서는 숫자를 7로 나누어 나머지가 0이 되어야 한다.’, ‘190의 약수는 190에 나누었을 때 나머지가 0이 되는 수이다.’ 등의 수학적 개념을 논리적으로 분석하고 작은 내용으로 분해하였고, 이를 절차적으로 정리하는 추상화 과정과 함께 알고리즘 사고, 절차적 사고, 반복적 사고, 비판적 사고, 논리적 사고를 관찰할 수 있었다.

나. 자동화

Wing(2006)은 추상화를 정신적 도구(mental tool)라고 한다면 이러한 추상화의 기능은 기계적 도구(metal tool)인 자동화를 통해 더 증폭되어 진다고 말한다. Wolfram(2020)은 컴퓨터를 활용하여 문제를 해결할 수 있는 역량에 주목하였는데, 학생들은 추상화하여 분해한 문제해결과정을 Python의 언어로 적용하는 과정인 코딩을 통해 자동화하는 과정을 경험하였다.

Wolfram(2020)은 컴퓨팅 사고의 네 단계(문제 정의 - 계산 가능한 형태로 추상화 - 계산 - 결과 해석)를 제시하며, 컴퓨터를 활용하여 문제를 해결할 수 있는 역량을 주목하였다. 그는 계산은 컴퓨터가 하게 하고, 인간은 다른 과정에 대한 활동에 좀 더 시간을 기울이는 것이 앞으로 다가오는 미래 사회를 대비할 수 있는 수학 학습의 핵심이라고 하였다.

[표 17]에서처럼 학생들은 수학적 문제해결을 돕기 위해 계산 대신 태블릿과 Python을 사용하여 학습한 명령어를 자동화하는 과정에서 학생들은 도전 의식을 보였고 성취감도 느꼈다. 또 S4와 같이 프로그래밍과 수학과의 관계를 인식하고 이전보다 수학에 대한 친밀감과 흥미를 더 갖는 모습을 관찰할 수 있었고, S9와 같이 문자를 랜덤으로 출력하는 방법을 물어보거나 S1과 같이 Python의 다른

명령어를 물어보는 등 더 많은 명령어를 학습하여 더 발전된 프로그램을 구현하고자 하는 의욕을 보여주기도 했다. 그리고 수학적 개념을 보고 느끼는데 그치지 않고 태블릿을 이용하여 자신의 수학 문제와 관련한 직접적인 조작이나 입력을 통하여 시뮬레이션해 봄으로써 주어진 문제에 대한 깊은 이해와 예측이 가능하게 되었다.

S9 : 선생님 저는 숫자 말고, 문자를 랜덤으로 출력하고 싶은데, 명령어 어떤 것을 쓰면 돼요?
 ... 다른 수학도 Python으로 해보고 싶어요.
 S1 : Python은 배우면 배울수록 알고 싶은 명령어가 많아요.
 더 배워서 더 많은 프로그램을 만들고 싶어요.
 S4 : Python에 수학이 있는 것이 신기했어요.
 수학과 프로그래밍에 대한 저의 마음이 조금 더 깊어진 것 같아요.

[표 17] 자동화에 관련된 학생 발화 및 소감문 예시

본 연구에서는 수업마다 자신의 프로그램을 코딩하고 구현한 결과를 바탕으로 오류를 수정하도록 하였다. 학생들이 프로그래밍 과정에서 디버깅하는 이유에 대해 생각해 보게 하고, 디버깅을 통해 더 발전된 프로그램을 만들어 보도록 하였다. 하나의 예시로 [그림 6]을 보면 S4는 첫 번째로 산술 연산 결과를 출력하기 위해 '='이 아닌 '=='으로 명령어를 수정하였고, 두 번째로는 문자를 출력하기 위해 “짝수”를 “짝수”로 작은 따옴표를 큰따옴표로 수정하여 통일하는 디버깅 과정을 관찰할 수 있다.

→ →

[그림 6] 오류를 수정하는 디버깅 과정 1

또 다른 예시로 [그림 7]을 보면 S12는 첫 번째 변수로 숫자를 입력하여 입력되는 숫자가 짝수이면 당첨, 홀수이면 낙첨으로 프로그래밍하였는데, 두 번째로 랜덤 함수를 이용하여 랜덤으로 출력되는 숫자가 짝수이면 당첨, 홀수이면 낙첨으로 변경하였다. 이는 수업 당시 S12가 교사에게 랜덤으로 출력된 숫자로 홀수와 짝수를 판단하고 싶다며 이야기하였는데, 게임 프로그램에 재미 요소를 더하기 위했던 것으로 판단된다. 마지막으로 랜덤으로 출력되는 숫자를 실행 창에 나타내기 위해 교사의 도움으로 'print(랜덤)'을 추가하는 등의 디버깅 과정을 관찰할 수 있다.

→

→

[그림 7] 오류를 수정하는 디버깅 과정 2

학생들이 디버깅하는 모습을 보면 화면을 쳐다보면서 거의 멈춘 것처럼 보이는 장면이 자주 수업 장면에서 나타났는데, 학생들은 문제해결을 위해 집중력을 유지하면서 디버깅 및 프로그래밍을 하는 것을 알 수 있었다.

또한 자신이 만든 프로그램을 공유하고 친구들이 비평해 보는 활동을 통해 자신의 프로그램에 대해 객관적으로 생각해보고, 더 나은 프로그램을 구현하기 위해 노력하려는 모습도 보였다. [표 18]은 S1이 게임을 만드는 과제를 해결하는 모습으로, 프로그램을 구현한 결과를 친구와 공유하는 과정에서 오류를 발견하고 수정하는 모습이 담겨있다. 추측, 반복, 시행착오와 같은 익숙한 문제해결 전략을 사용하며(신동조 외, 2019), 프로그램을 구현하는 것을 알 수 있다.

S1 : (친구에게) 난 이거 거의 다 만들었어. 몬스터 물리치기 게임이야.
한번 해 볼래? 몇 단계 할래?
(오류 발견) 잠시만, 일단 만들고 있는 중이야.
(수정 후) 다시 해 볼래?

[표 18] 프로그램 구현 결과 공유 장면

이처럼 학생들은 앞서 추상화한 해결 방법을 프로그래밍 단계와 생각해보기 단계에서 Python 프로그램으로 문제해결 단계를 코딩하고, 이를 구현하고 공유하는 과정에서 발견한 오류를 디버깅하는 체계적인 시뮬레이션을 실시하며 자동화 과정을 경험할 수 있었다.

VI. 결론 및 제언

본 연구에서는 융합 수업 프로그램에서 발현되는 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고를 분석하여 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고에 Python을 이용한 융합 수업 프로그램이 어떤 교육적 효과를 지니는지 알아보았고, 그 결과를 바탕으로 한 결론은 다음과 같다.

1. 초등 수학 영재의 융합 수업 프로그램

융합 수업 프로그램은 프로그래밍을 활용한 융합형 모델과 이야기·동화 모델을 적용하여 개발되었는데, 첫 번째로 프로그래밍을 활용하여 초등 수학 영재의 수학적 창의성과 함께 컴퓨팅 사고가 발현되는 것을 관찰하였다. 두 번째로 이야기·동화 모델을 적용하여 수학적 상황이 내재된 스토리텔링을 통해 수학적 상황을 발견하고 이를 이용하여 수학을 탐구하게 하였다.

본 연구에서는 Python 프로그램을 적용해 볼 수 있는 단계들을 먼저 구성하고 이에 따라서 수학적 창의성과 컴퓨팅 사고가 증진될 수 있는 수업 프로그램을 개발하였는데, 선행 연구들을 바탕으로 Polya의 문제해결 모형과 대응되는 4단계로 구분하여 수업을 구성하였다. 수업내용으로는 변수와 자료구조의 활용, 제어문(반복문, 조건문), 함수의 순서로 구성하였는데, 명령어의 단계적 교육을 통해 결과물을 더욱 발전시켜 나가는 방법을 반영하여 Python의 기초 문법 및 명령어들을 예제들을 통해 익히고, 실제적인 문제들(게임 만들기, 구구단 만들기, 최대공약수 구하기 등)을 해결해 보며 학습 참여와 성취도를 높일 수 있었다. 교육 주제는 프로그래밍의 난이도와 초등 영재 학생들의 수준을 고려하여 선정하였으며 총 9차시의 수업 프로그램을 설계하고 개발하였다.

2. 초등 수학 영재의 융합 수업 프로그램 효과 분석

가. 수학적 창의성

융합 수업 프로그램에서 나타난 수학 영재들의 수학적 창의성을 크게 수학적 창의성의 3수준과 수학적 창의성의 구성 요소인 인지적 요소, 정의적 요소, 태도적 요소로 분석하였고, 문제해결과정과 산출 결과에서 수학적 창의성을 관찰할 수 있었던 학생들을 중심으로 분석하고 기술하였다.

첫 번째로 수학적 창의성의 3수준으로 분석한 결과, 대체로 수업을 진행하면서 학생들은 점차 높은 수학적 창의 수준의 코드를 표현하는 경향을 보였다. 특히 4차시부터 학생들의 창의적인 사고가 뚜렷하게 관찰되기 시작했는데, 수학적 개념과 알고리즘, 절차 등에 대한 정확한 학습과 경험과 개방형 문제, 적당히 도전적이고 흥미로운 문제 제시, 활발한 수학적 의사소통이 수학적 창의성 발현에 도움이 되었을 것으로 판단된다.

두 번째 인지적 영역으로 학생들이 주어진 문제를 창의적으로 해결하기 위해 기존에 알고 있던 수학적 개념과 수학적 상징 기호, 프로그래밍 명령어 등을 자연스럽게 조직하고 형식화하여 프로그래밍하는 모습을 관찰할 수 있었다. 본 연구에서 수학적 창의성 발현을 위해 이용 가능한 정보로부터 새로운 변화를 만들

고 다양한 해결 과정을 도출할 수 있도록 문제를 설계하였으며, 일부 수학 영재들은 직관과 통찰을 통해 문제해결을 위한 단서를 지속해서 찾다가 명령어의 오류와 수학적 개념에 대해 생각이 미치자 순간적으로 해법을 찾았고, 문제해결 및 디버깅 과정에서 아이디어를 정련하고 자신의 사고를 점검하고 반성하는 사고를 관찰할 수 있었는데, 이러한 사고 과정과 기술이 새롭고 다양한 산출물 도출에 도움이 된 것을 알 수 있었다.

세 번째 정의적 영역으로 창의적인 코딩 결과를 도출한 학생과 수업 후 면담 과정에서 Python을 활용한 본 수업에 대한 흥미를 느꼈고, 스스로 다르게 프로그래밍하고 싶은 내적 동기를 관찰할 수 있었다. 유창성을 보이며 다양한 프로그램을 제시한 학생은 수업 중간에 개인적으로 연구자에게 궁금한 것에 대해 질문을 꾸준히 하며 수업내용에 대한 호기심을 보이곤 하였는데, 본 연구의 수업을 통해 수학 영재들은 새로운 수학에 대한 탐구와 열정을 가지게 되며, 이러한 성향이 더 창의적인 프로그램 결과를 도출하고자 노력하는 원동력이 된 것으로 사료된다.

마지막은 태도적 영역으로, 독창적인 게임 프로그램을 완성한 학생이나 직관과 통찰을 바탕으로 새로운 프로그램을 제시한 학생들에게서는 실제로 문제를 창의적으로 해결하는 과정에서 과제에 대한 몰입과 타인의 간섭이나 평가에 구속되지 않는 독립심, 새로운 문제에 도전하려는 도전 의식과 그에 따른 성취감 등을 관찰할 수 있었다. 이러한 과제집착력, 독립심, 도전감 등이 외적인 간섭과 평가에서 벗어나 자신만의 아이디어를 산출하는 데 관계가 있을 것으로 판단된다.

나. 컴퓨팅 사고

융합 수업 프로그램에서 나타난 수학 영재들의 컴퓨팅 사고를 크게 추상화, 자동화(Wing, 2006)를 중심으로 분석하였고, 문제해결과정과 산출 결과에서 컴퓨팅 사고를 관찰할 수 있었던 학생들을 중심으로 분석하고 기술하였다.

첫 번째로 수학 영재들은 각 차시마다 자신의 활동을 필기구를 이용하여 알고리즘으로 표현하며 손코딩 하는 활동에 참여하였다. 또한 절차적 사고를 거쳐 불필요한 요소를 제거하고 핵심 요소를 추출하는 활동과 수학적 개념을 구체적으로 분석하고 작은 내용으로 분해하는 활동을 경험하였고, 스토리텔링 속 핵심 요소를 인지하여 전체 과정을 순서대로 정리하는 추상화 과정과 함께 문제 해결의 순서를 생각하고 적용하는 절차적 사고(Procedural thinking), 최대공약수를 구하는 과정에서 두 개의 임시 변수를 사용하여 반복적으로 연산하는 반복적 사고, 최대공약수를 구하는 표현 방법들 중 하나의 방법을 선택하여 정확한 코드로 표현하는 일종의 알고리즘 사고(Algorithmic thinking)와 비판적 사고(Critical

thinking)를 관찰할 수 있었다. 문제해결과정에서도 알고리즘 사고와 함께 모델링 과정을 관찰할 수 있었는데, 스스로 핵심 요소를 찾고 필요한 과정만 표현하며 순서대로 정리하고 손코딩 하였으며, 수업 후 면담에서 자신이 프로그램을 추상화한 과정과 디버깅한 사실을 논리적이고 정확하게 연구자에게 설명하였다.

두 번째로 수학 영재들은 앞서 추상화한 해결 방법을 프로그래밍 단계와 생각해보기 단계에서 Python 프로그램으로 문제해결 단계를 코딩하고, 이를 구현하고 공유하는 과정에서 발견한 오류를 디버깅하는 과정에서 자동화 능력을 관찰할 수 있었다. 수학 영재들은 수학적 문제해결을 돕기 위해 태블릿과 Python을 사용하여 학습한 명령어를 자동화하는 과정에서 성취감과 도전 의식을 보여주었으며, 더 많은 명령어를 학습하여 더 발전된 프로그램을 산출하고자 하는 의욕을 보여주기도 했다. 학생들이 프로그래밍 과정에서 디버깅하는 이유에 대해 생각해 보게 하고, 디버깅을 통해 더 정확하고 발전된 프로그램을 만들어 보도록 하였다. 학생들이 디버깅하는 모습을 보면 거의 멈춘 것처럼 보이는 장면이 수업 장면에서 나타났는데, 학생들은 문제해결을 위해 집중력을 유지하면서 디버깅 및 프로그래밍을 하는 것을 알 수 있었다. 또한 자신이 만든 프로그램을 공유하고 친구들이 비평해 보는 활동을 통해서 익숙한 문제해결 전략(추측, 반복, 시행착오 등)을 사용하며 문제해결 과정을 자동화하였다.

3. 논의 및 제언

본 연구는 J초등학교 수학·과학 영재학급 학생들을 대상으로 프로그래밍을 활용한 융합 수업 프로그램을 적용하고, 그 과정에서 나타나는 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고를 분석하여 초등 수학 영재들의 수학적 창의성과 컴퓨팅 사고에 현대적 프로그래밍 언어인 Python 프로그래밍을 이용한 융합 수업 프로그램이 어떤 교육적 효과를 지니는지 알아보려고 하였다. 따라서 연구 결과를 토대로 프로그래밍을 활용한 융합 수업에 관한 시사점을 아래와 같이 논의하고, 연구 과정에서의 제한점을 토대로 후속 연구에 대하여 제언하고자 한다.

첫째, 본 연구에서 제시한 융합 수업 프로그램은 초등 수학 영재들을 위한 4차 산업혁명 시대에 적합한 융합 교육 프로그램으로 활용될 수 있고, Python을 이용하여 창의적으로 코드를 표현하고 프로그램을 완성하는 과정에서 학습자들이 수학적 개념과 사고, 프로그래밍을 융합적으로 탐구할 수 있도록 설계하였다는 점에서 의의가 있다.

둘째, 융합 수업 프로그램이 효율적으로 이루어질 수 있도록 학습 내용을 보다 체계화하고 정당성을 더욱 확보하는 연구가 필요하다. 본 연구에서 제시한 융합 수업 내용은 선행 연구와 전문가 검토를 바탕으로 개발되었으나, 16명의 영재학

급 단일집단에 적용하여 그 결과를 분석하였기 때문에 연구 결과를 일반화하기 어려울 수 있다. 또한 수학 영재들이라 할지라도 인지적인 수준이 다양함을 고려할 때, 초등 수학 영재들을 위한 프로그래밍을 활용한 융합 수업의 체계화가 필요하다(장인옥, 2010). 따라서 프로그래밍 교육과정에 적합한 수업 모형이나 내용 구성을 더 연구하여 교수 방법을 정립하고, 전문가 집단의 설문을 추가하여 시행하며 그 결과를 반영한다면 본 연구의 융합 수업의 체계화와 정당성을 더욱 확보할 수 있을 것이다.

셋째, Python 프로그래밍 과정에서 수학적 창의성이 발현될 수 있는 개방형 과제와 장기적인 수업 구성에 대한 연구가 필요하다. 본 연구의 융합 수업은 다소 짧은 기간이라 할 수 있는 9차시로 구성되었기 때문에 제한적인 프로그래밍 지식으로 해결할 수 있기 위해 일부 닫혀있는 과제를 제시하였는데, 이러한 점이 수학 영재들의 보다 높은 수준의 수학적 창의성 발현에는 한계가 있었다. 따라서 후속 연구에서는 장기적인 수업 내용을 바탕으로 다양한 개방형 과제를 제시하는 등, 수학 영재들의 높은 수준의 수학적 창의성 발현을 위해 보다 효율적이고 체계적인 방안이 모색되어야 할 것이다.

넷째, 수학 영재들에게 Python 프로그래밍 문제를 제시할 때, 연구대상자의 수준에 적합하면서 도전적이고 흥미로운 과제를 모색할 필요가 있다. 학생들은 좋아하는 게임 프로그램을 완성하는 수업에서 수업 내용에 대한 호기심을 보이며, 높은 수학적 창의성을 발휘하는 모습을 관찰할 수 있었다. 또한 마지막 차시의 내용인 함수, 반복문, 조건문과 관련된 코딩 명령어를 함께 사용해서 최대공약수를 구하는 프로그램을 만드는 과정에서 유클리드 호제법에 관심을 보이며, 새로 알게 된 수학적 지식에 대한 만족감을 드러내었다. 그리고 지적 호기심으로 어려운 문제에 도전 의식을 가지고, 학습한 수학적 개념과 명령어로 표현하여 형식화하는 모습도 관찰할 수 있었다. 따라서 후속 연구에서는 학생들의 수준과 흥미를 고려한 좀 더 다양하고 도전적인 프로그래밍 과제가 모색되어야 할 것이다.

다섯째, 본 연구의 융합 수업 프로그램을 일반 학생에게 적용하여 수학 영재과의 문제해결 과정 및 결과를 비교하는 연구가 필요하다. 선행 연구에서 영재 학생과 일반 학생의 문제해결 과정 및 결과를 비교한 연구들이 제시되고 있지만(전영국, 2015), 프로그래밍을 활용한 융합 수업을 적용하여 그 결과를 비교한 연구는 미흡한 실정이다. 따라서 후속 연구에서 일반 학생과 영재 학생의 프로그래밍 과정과 결과를 비교하는 연구를 실시한다면, 학생의 수준과 특성에 맞게 수학적 창의성과 컴퓨팅 사고를 효과적으로 개발할 수 있는 융합 수업 프로그램을 개발하기 위한 기초 자료가 될 것이다.

마지막으로, 본 연구는 융합 수업 프로그램을 통해 초등 수학 영재들의 수학적

창의성을 관찰하고 분석하는 것을 중심으로 하였기 때문에, 초등 수학 영재들의 컴퓨팅 사고를 심도 있게 분석하지 못한 점에 대해 아쉬움이 있다. 따라서 향후 본 연구의 수업에 컴퓨팅 사고를 분석하였던 도구들을 더 세분화하고 보완하여 프로그래밍 수업의 핵심 목적인 컴퓨팅 사고를 보다 심층적으로 분석하는 것도, 초등 수학 영재 교육에 또 다른 시사점을 제시할 수 있는 의미 있는 연구가 될 것이다.

본 연구에서 개발된 융합 수업 프로그램이 초등 수학 영재를 위한 융합 및 프로그래밍 관련 사례 연구에 있어서 창의적 사고와 컴퓨팅 사고의 도구로 유용하게 사용되며 후속 연구들로 이어질 수 있기를 기대한다.

참고문헌

- 강신천, 안성진, 성영훈, 정영식, 김영애, 서정희, 박세영 (2019). 해외 소프트웨어교육 운영 현황에 대한 실증자료 분석 리포트. KERIS 이슈 리포트, 2019(3).
- 강주영, 김동화 (2020). 중등 수학 영재의 독창성에 관한 고찰. 과학영재교육, 12(3), 212-225.
- 교육부 (2015a). 수학과 교육과정. 교육부 고시 제 2015-74호 [별책8번].
- 교육부 (2015b). 제2차 수학교육 종합 계획. 교육부 보도자료(2015.03.16.).
- 권오남, 박정숙, 박지현, 조영미 (2005). 시리즈 A: 개방형 문제 중심의 프로그램이 수학적 창의력에 미치는 효과. 한국수학교육학회지 시리즈 A-수학교육, 44(2), 307-323.
- 김나리, 서용현, 조한혁 (2018). 코딩수학 내용 및 환경 설계-수학화와 컴퓨팅 사고력을 중심으로-. 학습자중심교과교육연구, 18(4), 647-673.
- 김동화, 김영아, 강주영 (2018). 중등수학영재의 수학적 창의성에 대한 고찰. *East Asian Mathematical Journal*, 34(4), 429-449.
- 김병수 (2014). 계산적 사고력 신장을 위한 PPS기반 프로그래밍 교육 프로그램. 제주대학교 대학원 박사 학위논문.
- 김부윤, 이지성 (2007). 수학적 창의성에 대한 관점 연구. 수학교육, 46(3), 293-302.
- 김상룡 (2002). 초등수학에서 동화의 활용 방안 탐색. 초등수학교육, 6(1), 29-40.
- 김언주 (1997). 창의력 사고에 있어서 정서의 문제. 창의력교육연구, 1(2), 171-189.
- 김영직 (2020). 문제해결 프로그래밍 교육을 위한 컴퓨팅 사고력 기반 테스트 중심 문제해결(CT-TDPS) 학습 모형 개발 및 적용. 한국교원대학교 대학원 박사학위논문.
- 김예미, 고호경, 허난 (2020). 파이썬을 활용한 중학교 1학년 소인수분해의 수학과 코딩 융합 교수-학습 자료 개발 연구. 수학교육논문집, 34(4), 563-585.
- 김원준, 김은아, 오세경 (2019). 유아코딩교육에 대한 유아교사의 인식분석. 열린유아교육

- 연구, 24(2), 307-337.
- 김유경, 방정숙 (2015). 수학 기반 융합 수업 모형의 가능성 탐색. *初等 數學教育*, 18(2), 107-122.
- 김종진 (2011). EPL을 이용한 창의성 증진 교육 프로그램 개발 및 적용에 관한 연구. 홍익대학교 대학원 박사학위논문.
- 김판수 (2008). 창의성 이론을 통해 본 수학 창의성. *영재교육연구*, 18(3), 465-496.
- 김홍원, 박경미, 김수환, 김신영, 채선희 (1997). 창의력 신장을 돕는 중학교 수학과 학습 평가 방법 연구. 한국교육개발원 연구보고서.
- 남승인, 류성림, 신준식, 우동하, 이용희 (2016). 수학영재교육: 이론과 실제. 경문사.
- 류성림, 이종학, 윤마병, 김학성 (2018). 초등학교 영재교육을 위한 수학·과학 중심의 융합 교육 프로그램 개발. *한국융합학회논문지*, 9(10), 217-228.
- 류희찬, 장인옥 (2010). LOGO를 이용한 프로젝트 학습에서 나타난 초등 수학영재 학생들의 전략적 사고. *대한수학교육학회지 수학교육학연구*, 20(4), 459-476.
- 문성재, 노정원, 노예솔, 이경화(2019). 연구공동체 활동을 통한 한 경력교사의 전문성 신장: 수학적 창의성 촉진을 위한 대푯값 과제의 변형과 실행을 중심으로. *학교수학*, 58(4), 545-566.
- 문용린 (2010). 배려와 나눔을 실현하는 창의인재육성을 위한 창의·인성교육 활성화 방안 연구. 한국과학창의재단.
- 박대륜, 유인환 (2018). 초등학생을 위한 로봇 활용 파이썬 학습 모형 개발. *정보교육학회논문지*, 22(3), 357-366.
- 박성현 (2002). 컴퓨터과학을 위한 수학교육. 중앙대학교 석사학위논문.
- 박재현 (2014). 평이한 정형문제를 통한 창의성 수준과 성향에 대한 사례연구. 한국교원대학교 대학원 석사학위논문.
- 박주연 (2015). Scratch 프로그래밍 수업에서 학습자 특성, 학습몰입, 학습 효과의 구조적 관계 규명. 이화여자대학교 대학원 박사학위논문.
- 서용현 (2019). 초중등 연계 코딩수학 교육과정 개발 및 적용. 서울대학교 교육대학원 석사학위논문.
- 신동조, 고상숙 (2019). 수학교육에서 계산적 사고(Computational Thinking)의 의미 및 연구 동향 탐색. *수학교육*, 58(4), 483-505.
- 신수범 (2015). 스크래치 소프트웨어 교육을 통한 컴퓨팅 사고력 향상 효과. *한국컴퓨터정보학회논문지*, 20(11), 191-197.
- 심광섭, 심성아 (2018). 파이썬 코딩을 도입한 수학 교과 지도 방안 개발-2015 개정 교육과정 중학교 수학 교과의 '소인수분해' 내용을 중심으로-. *성신여자대학교 교육문제연구소*, 73, 43-64.
- 안성진 (2015). Computational Thinking에 대한 이해, 정보문화포럼 2015년 연구보고서.

- 양재명, 이원규, 김자미, 윤일규, 서정희, 우호성, 양혜지, 김민정, 최희정 (2017). 2017년도 소프트웨어(SW)교육 연구학교 현황 및 효과성 분석. 한국교육학술정보원.
- 유정호 (2015). 수학 내용 기반의 코딩 교육 프로그램 개발. 서울교육대학교 교육전문대학원 석사학위논문.
- 유중현, 김종혜 (2008). 문제해결과정에서의 정보과학적 사고 능력에 대한 개념적 고찰. 정보창의교육논문지, 2(2), 15-24.
- 이경화 (2015). 수학적 창의성. 경문사.
- 이대현 (2012). 수학적 창의성의 요소와 창의성 개발을 위한 수업 모델 탐색. 한국초등수학교육학회지, 16(1), 39-61.
- 이도영, 정종인 (2018). 중학교 수학 통계 영역과 파이썬(Python) 프로그래밍 융합수업이 문제해결력과 교과 흥미도에 미치는 영향. 한국산학기술학회논문지, 20(4), 336-344.
- 이승우 (2020). 프랑스 중학교 수학 교육과정 분석: '알고리즘과 프로그래밍' 영역을 중심으로. 학교수학, 22(1), 125-159.
- 이은석 (2013). LOGO에서 별다각형 그리기를 활용한 초등수학영재교육 학습 지도 자료 개발. 공주교육대학교 교육대학원 석사학위 논문.
- 이재호, 장준형 (2018). 컴퓨팅 사고력 검사도구 개발을 위한 탐색. 창의정보문화연구, 4(3), 273-283.
- 이재호, 장준형 (2020). 일반학생과 영재학생의 소프트웨어교육을 통한 컴퓨팅 사고력 신장 효과 비교 분석. 영재교육연구, 30(1), 55-64.
- 이지성 (2006). 수학적 창의성의 창의적 태도에 대한 측정 도구의 개발과 적용. 부산대학교 대학원 박사학위논문.
- 장인옥 (2010). LOGO를 이용한 프로젝트 학습에서 나타난 초등 수학 영재 학생들의 전략적 사고와 교사 역할. 한국교원대학교 대학원 박사학위논문.
- 전영국 (2015). 컴퓨터 프로그래밍과 창의성 발현 활동에 관한 질적 사례 연구: NetLogo 기반의 계산적 사고 중심으로. 컴퓨터교육학회 논문지, 18(3), 1-14.
- 정인우, 조한혁 (2020). 3차원 좌표계 기반 코딩환경을 활용한 수학적 창의성 발현 방안: 코딩과제 설계 및 코드표현의 분석을 중심으로. 학교수학, 22(1), 161-181.
- 정진주 (2020). 컴퓨팅 사고력 신장을 위한 초등 수학 영재용 학습 자료 개발. 경인교육대학교 교육전문대학원 석사학위논문.
- 지현경 (2021). 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형 개발 연구. 서울대학교 대학원 박사학위논문.
- 최인선 (2012). 수학적 창의성 연구에 대한 고찰. 한국수학교육학회 학술발표논문집, 2012(2), 29-32.
- 황우형, 최계현, 김경미, 이명희 (2006). 수학교육과 수학적 창의성. 한국수학교육학회지

- 시리즈 E 수학교육 논문집, 20(4), 561-574.
- 황혜진 (2016). 스크래치를 활용한 방과후 융합인재교육(STEAM) 수학수업 교육과정 개발 및 적용. 신라대학교 교육대학원 석사학위논문.
- Becker, J. P., & Shimada, S. (1997). The open-ended approach: A new proposal for teaching mathematics. Reston, VA: NCTM.
- Clement, D. H., Battista, M. T., & Sarama, J. (2001). Logo and geometry. *Journal for Research in Mathematics Education, Monograph, 10*, i+1-177.
- Ervynck, G. (1991). *Mathematical creativity*. In D. Tall. (Ed.), *Advanced mathematical thinking* (pp. 42-53). Netherlands: Kluwer Academic Publishers.
- Haylock, D. W. (1987). A framework for assessing mathematical creativity in schoolchildren. *Educational Studies in Mathematics, 18*(1), 59-74.
- CSTA & ISTE. (2011). *Operational definition of computational thinking for K-12education*. Retrieved June 17, 2019, from <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- Krutetskii, V. A. (2014). 수학적 능력의 심리학 [송상헌, 임재훈, 권석일, 남진영, 정영옥, 서동엽, 김성준, 김지원, 역] (원본 출간년도 : 1976). 서울 : 경문사.
- Lee, K. H. (2017). Convergent and divergent thinking in task modification: A case of Korean prospective mathematics teachers' exploration. *ZDM, 49*(7), 995-1008.
- Polya, G. (1957). *How to solve it: A new aspect of mathematical method(2nd)*. (Ed.), 우정호 (역) (2008). 어떻게 문제를 풀 것인가. 교우사.
- Silver, E. A. (1997). Fostering Creativity through Instruction Rich in Mathematical Problem Solving and Problem Posing. *ZDM, 29*(3), 75-80.
- Sriraman, B. (2005). Are giftedness and creativity synonyms in mathematics? *The Journal of Secondary Gifted Education, 17*(1), 20-36.
- Tall, D. (1991). *Advanced mathematical thinking*. 류희찬, 조완영, 김인수(역)(2003). 고등 수학적 사고. 경문사.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology, 25*(2), 7-14.
- Wolfram. C. (2020). *The Math(s) Fix: An Education Blueprint for the AI Age-Wolfram Media*. Wolfram Media, Inc.

Kang, Joo Young
JIN-YEONG GEUMBYEONG Elementary School
Gimhae, 50862 Korea
E-mail address: 14292701@naver.com

Kim, Dong Hwa
Department of Mathematics Education
Pusan National University
Pusan, 46241 Korea
E-mail address: dhgim@pusan.ac.kr

Seo, Hae Ae
Department of Biology Education
Pusan National University
Pusan, 46241 Korea
E-mail address: haseo@pusan.ac.kr