

하노이 탑 프로그래밍 경험에서 나타나는 정보과학적 사고 패턴에 관한 질적 사례 연구

장정숙[†] · 전영국^{††} · 윤지현^{†††}

요 약

본 연구는 고등학생이 컴퓨터 프로그래밍을 사용하여 문제를 해결할 때 아이디어를 어떻게 포착하며 아이디어를 어떻게 구체화시키면서 코딩을 해 나가는지를 살펴봄으로써 다양한 논리적 사고력의 발현과 오류 수정을 통해 정보과학적 사고와 관련된 능력이 펼쳐지는 양상을 탐구하였다. 연구참여자 K를 대상으로 2011년 10월부터 5회에 걸쳐 비디오 기반 회상 면담과 심층면담을 혼합하여 하노이탑 과제에 대한 C 프로그래밍을 하는 경험을 데이터로 수집하였다. K는 컴퓨터 프로그래밍을 할 때 ‘처음에 문제를 천천히, 차근차근 읽어보고 직접 노트에 쓰기도 하고, 그 문제를 분석해서 패턴들을 찾아내고 생각을 구체화하여 알고리즘을 적용하고 컴퓨터로 구현해 나가는 것’ 방식을 드러내 보였다. 이러한 과정에서 ‘자기만의 생각을 실험해 보는 것에’ 재미를 느끼면서 과학적 사고에 가까운 사고 패턴을 보여주었으며 일상 생활에서 알고리즘적 사고를 적용하는 등 사고력의 향상에 관련되는 부분을 드러내 보여 주었다.

주제어 : 프로그래밍 교육, 정보과학적 사고, 질적 사례 연구

Qualitative Case Study on Computational Thinking Patterns of Programming Processes for the Tower of Hanoi Task

Jeongsuk Jang[†] · Youngcook Jun^{††} · Jihyun Yoon^{†††}

ABSTRACT

This paper aims at exploring how a high school student is engaged with C computer programming language and explore deep aspects of programming experiences based on video recalled interview as part of portraiture. The single case was selected and several in-depth interviews and video recording were arranged after Oct 2011. The portrait of K reveals interesting strands of his computer programming experiences with his own thinking patterns and exploring ideas for more concrete ways of coding his thinking similar to scientific experiments; ie, design-development-implementation-debugging-revision. Overall, this case illustrates how the inner aspects of subjective programming experiences on the tower of Hanoi were integrated and unified within himself for inner growth. We discussed the student's inner faculties as part of the student's unity and suggested future research direction.

Keywords : Programming Education, Scientific Thinking, Qualitative Case Study

† 정 회 원: 순천대학교 컴퓨터교육과

†† 중신회원: 순천대학교 컴퓨터교육과 교수(교신저자)

††† 준회원: 순천대학교 과학정보융합학과

논문접수: 2013년 06월 11일, 심사완료: 2013년 07월 25일, 게재확정: 2013년 07월 29일

* 본 논문은 2013년 1월 동계 학술대회에서 우수 논문으로 선정된 내용을 토대로 수정 보완하여 작성되었습니다.

1. 서론

최근 개정된 교육과정은 지식 정보 사회에서 유용하게 사용되는 정보통신 기술을 효과적으로 사용하여 문제 해결력을 기르는데 목적을 두고 있다[1]. 문제해결능력을 향상을 위해서는 정보과학 분야의 계산적 사고(computational thinking)가 요구되어진다. 컴퓨터 프로그래밍은 자신만의 아이디어를 바탕으로 주어진 문제를 분석하고, 해결하는 일련의 과정들을 반복하면서 끊임없이 분석하고 생각하고 실험한 끝에 결과를 얻게 된다[9]. 이러한 컴퓨팅의 매력은 논리적 사고력을 기반으로 자신의 아이디어를 창의적으로 구현할 수 있는 프로그래밍 공간에서 코딩 실력을 쌓아가면서 한 단계 한 단계 성장해 나가고 나아가 실생활에 까지 적용가능하다.

이러한 분야의 주제를 다루는데 질적 연구 방법이 활발히 도입되고 있다. 특히 최근에 미디어 및 로봇 활용 교육이 활성화 되면서 교육 실천가들은 수업 관찰 및 학생 활동에 대한 과정을 분석할 필요성이 많아지고 있는 가운데 교실 수업을 관찰하기 위하여 비디오 촬영과 면담을 통한 자료 수집에 의존하는 경향을 보이고 있다. 예를 들어 미디어 활용 수업, 프로그래밍 수업 또는 방과후 로봇수업에 참여하는 학생들의 활동을 비디오로 촬영한 후에 분석함으로써 매체(멀티미디어, 컴퓨터 프로그래밍, 로봇 키트)를 다루는 학생들의 특성과 활동 과정에 대한 미세한 부분을 탐색할 수 있다[3][20]. 특히 미디어를 활용하여 창의적인 활동을 지원하는 수업에 대한 연구를 하는데 비디오 촬영과 비디오 회상 면담 방법이 유용하게 사용될 수 있다. 예를 들어, 프로그래밍을 공부하면서 정해진 목표에 따라 효율적으로 공부하는 생활방식은 컴퓨팅 작업을 통하여 논리적이고 명확하게 사고하는 방식과 잘 부합하고 있다 [2]. 그리고 컴퓨터 프로그래밍 교육 및 알고리즘은 논리적 사고력이나 문제해결력, 창의력에 긍정적인 영향을 준다[4][5][6].

본 연구에서는 문제해결 과정에서 나타나는 논리적, 비판적, 재귀적, 알고리즘적 사고와 같은 정보과학적 사고가 실제 컴퓨터 프로그래밍 과정에서 어떻게 나타나고 있는지 비디오 촬영과 비디오

회상 면담을 통해 질적 사례 연구로 살펴보고자 한다[2][7][20].

2. 이론적 배경

컴퓨터 프로그래밍 과정을 살펴보면 프로그래밍 문제의 목표를 인식하고 계획을 세우며 프로그래밍 언어로 코딩하고, 오류수정을 통하여 목표와 현 상태의 차이를 최소화하는 과정을 거친다 [5][20]. 이와 비슷하게 프로그래밍 과정을 지식획득, 이해하기, 만들기, 교정하기, 수정하기로 설명하고 있다. 이러한 과정을 통해 이루어지는 프로그래밍 학습은 문제 해결능력을 신장시켜 주는 과정에서 과학적 사고와 밀접한 관련성을 띠고 있다. 질적 연구 방법 중에서 근거 이론을 사용하면 이런 현상에 대하여 이론적 틀을 구축해 나갈 수 있다[21].

본 연구에서는 컴퓨터 프로그래밍을 하는 참여자의 주관적 경험 속에서 과학적 사고 패턴과 관련된 요인과 연구참여자 내면 속에서 지속적으로 드러나는 주요 양상을 포착함으로써 프로그래밍 경험이 내적 성장과 어떤 관련성을 갖는지에 대하여 미시적으로 살펴보고자 한다[2]. 이를 위하여 컴퓨터 프로그래밍 절차를 문제 분석 단계 → 설계 단계 → 프로그램의 구현 단계 → 프로그램의 문서화 단계로 구분하였다.

컴퓨터 프로그래밍과 관련된 논리적 사고는 절차적 사고와 논리적 추론과 연관성이 깊다. 과학적 사고는 일반적으로 원인-결과의 인과적 관계를 다루며, 강력한 이론, 치밀한 분석, 논리적 증명, 타당한 주장, 가설로부터의 경험적 발견 등의 특징적 형태이다. 이러한 과학적 사고는 절차적 특징을 통해서 드러난다고 할 수 있으며 이러한 측면에서 컴퓨터 프로그래밍을 도입하여 문제 해결 능력을 배양하는 필요성이 지속적으로 제기되고 있다[4].

2.1 정보과학적 사고

CS4FN(Computer Science For Fun)은 컴퓨터를 활용하여 문제를 해결하는 다양한 기술들을 정보과학적 사고라고 하였다. 정보과학적 사고는

컴퓨터 과학을 다양한 분야에 적용시키기 위한 도구로서 복잡한 문제나 시스템 디자인을 할 때 사용하는 추상적 사고, 분할적 사고, 문제 해결의 예방책 등도 포함된다[8]. 정보과학적 사고를 세분화하여 살펴보면 다음과 같이 나눌 수 있다.

첫째, 알고리즘적 사고(algorithmic thinking)는 문제 해결에 필요한 과정을 순차적으로 나열, 적용하는 사고과정으로 결과 보다는 과정에 중점을 두고 어떻게 하는지에 중점을 두는 문제 해결을 위한 사고 과정이다[9]. 이러한 사고 과정에서는 문제 해결을 위해서 어떠한 과정을 따라야 한다는 직관적 사고가 필요하고 직관적 사고에 의해서 각각의 해결책들의 장단점을 파악하는 비판적 사고가 이미 포함되어 있다.

둘째, 재귀적 사고(recursive thinking)는 하나의 사고과정을 계속적으로 반복하여 사고하는 것으로 반복적인 작업을 통해 자료와 코드를 서로 바꿔주고 오류를 찾아 수정하며 해결책이 적절성, 효율성 그리고 간결성을 판단한다.

셋째, 비판적 사고(critical thinking)는 어떤 주장을 깊이 있고 폭넓은 맥락에서 잘 이해하려는 것으로 적극적으로 분석하고 종합하며 평가하는 능동적 사고이며 체계적이고 조직적인 사고이다.

넷째, 논리적 사고(logical thinking)는 미리 가지고 있는 정보를 사용하여 새로운 정보를 찾아내는 추론의 과정으로 논리적 사고가 성립하기 위해서는 비판적 사고가 뒷받침 되어야 한다. 실제 다양한 분야에 필요한 사고는 크고 복잡한 문제를 설계할 때 추상적 사고나 분할적 사고를 필요로 하며 해결책을 찾기 위한 귀납적 추론을 통해 결과를 유추하면서 새로운 해결방안을 계획하고 결과물을 찾아내는 문제해결전략을 사용하게 된다.

2.2 질적 사례 연구

인물 사례 연구를 진행하기 위하여 주제(예, 컴퓨터 프로그래밍 경험과 정보과학적 사고와의 관련성을 탐구)에 부합되는 연구참여자를 물색하여 연구 주제와 목적 등을 알려주고 동반자적 관계 형성을 한다. 주제에 부합되는 질문을 어떻게 구성하고 적용할 것인지를 생각하여 질문을 반구조

화하여 기본 면담을 1회 진행한다. 이러한 면담 자료를 바탕으로 좀 더 개방적인 심층면담을 진행하기 위해 첫 번째 면담 자료를 충분히 숙지한 후에 후속질문거리를 준비하며 연속적인 후속 면담에 있어서 상호적이고 즉각적인 반응이 나올 수 있도록 심층면담을 반발짝 정도 떨어진 상태에서 대화를 이끌면서 연구참여자가 스스로 이야기를 활발히 할 수 있도록 안내한다. 본 사례 연구에서 사용하였던 심층 면담의 목적은 연구주제와 관련하여 연구참여자의 내면 속에 면밀히 흐르는 주요 특성(strands)을 생생한 경험으로부터 포착함으로써 프로그래밍에 관련된 통합적 이해를 도모하는 것이다[2][16][17].

비디오 촬영은 이 연구와 같이 면담을 통해서 나타나지 않는 사례자의 비언어적 표현과 컴퓨터 프로그래밍의 과정에서 사례자만의 사고의 특징이 어떻게 나타나는지 다각적으로 이해하기 위해 필요하다. 비디오 카메라는 인간의 감각이 찾을 수 없는 여러 가지 교실들의 이미지들 그리고 상호작용들을 포착해 주고 나아가 보관하기 때문에 자료의 기술과 분석에 직접적으로 도움을 준다. 대화의 연속, 방해, 목소리의 억양, 속도, 그리고 참여자들의 표정 등이 비디오 카메라를 통하여 보존된다[10].

회상 인터뷰는 촬영된 비디오 자료를 보면서 비디오 촬영 중에 놓쳤던 의문점들을 질문하고 연구자가 잘못 해석한 부분이 있는지를 점검하고 정교화하기 위해 실시한다. 직접해석은 사례연구에서 이루어지는 해석의 방법으로 연구자는 여러 사례를 찾지 않고 단일 사례를 검토하여 그것으로부터 의미를 이끌어 내는 것이다[11][12].

프로그래밍 교육과 관련된 질적 사례 연구로는 Logo 언어를 다루는 학생을 대상으로 비디오 촬영 및 면담을 진행한 연구[11]와 Visual Basic 언어를 사용하여 문제해결을 다루는 정보영재반 학생을 대상으로 진행한 연구[20]가 있다. 문제해결 과정에서 나타나는 사고의 전개 과정과 문제해결의 단서를 찾아서 발전시켜 나가는 과정을 비디오 촬영과 면담을 통하여 연구를 진행함으로써 기존 이론과 같이 수집된 자료를 통하여 연구참여자의 프로그래밍 작업을 통합적으로 이해하는 특성을 보여주고 있다[21].

한편, 대학생들을 대상으로 컴퓨터교육을 전공하는 동기와 학업과 관련된 부분을 살펴본 연구를 살펴보면 중고등학교 시절에 컴퓨터에 관한 관심과 흥미를 점차 발전시켜 컴퓨터교육과에 입학한 경우가 많지 않음을 알 수 있다[19]. 그러나 그 반대의 경우에 컴퓨터 하드웨어와 소프트웨어를 공부하면서 점차 컴퓨터 전공 과목에 깊은 흥미를 가지고 소프트웨어 개발자의 길로 나아간 사례가 있다. 컴퓨터예비교사 1명을 대상으로 진행한 이 연구는 중고등학교 시절에 가졌던 컴퓨팅에 관한 관심과 흥미가 어떻게 점차 발전되어 나갔으며 컴퓨터 조립과 프로그래밍 언어를 배우는 과정에서 ‘나만의 것’을 추구하고자 하는 개인의 본질적인 측면을 마치 초상을 그리듯이 보여주고 있다[2].

3. 연구 설계

본 연구에서는 하노이 탑 과제 해결과정을 담은 비디오 자료와 과제 해결 당시의 컴퓨터 화면 캡처자료, 비디오 회상 면담 자료를 바탕으로 문제를 해결하는 학생에게 정보과학적 사고가 어떻게 나타나고 있는지 살펴보고자 한다.

3.1 비디오 분석 방법

교실 수업을 관찰하기 위하여 관찰 노트를 작성하는 방법은 면담 다음으로 많이 사용되고 있다. 예를 들어 미디어 활용 수업, 컴퓨터 프로그래밍 수업, 영재교육의 수업 또는 방과후 로봇수업에 참여하는 학생들의 활동을 비디오로 촬영한 후에 분석함으로써 매체(멀티미디어, 컴퓨터 프로그래밍, 로봇 키트)를 다루는 학생들의 특성과 활동 과정에 대한 미세한 부분을 탐색할 수 있다 [11][13][14].

비디오 화면에 담겨있는 자료 비디오 자료를 코딩하는데, 여기서 코딩이란 연구자가 비디오 내용을 검토하면서 의미있다고 판단되는 범주를 의미한다. 비디오 내용을 범주별로 코딩을 하고 나면, 전체 비디오 테이프에 담긴 내용을 일목요연하게 파악할 수 있는데, 가능하면 비디오에 담긴 오디오 내용을 전사한다. 이것을 토대로 전체 내

용에 대하여 선별적으로 깊이 있게 들여야 볼 대상을 좁혀나간다. 비디오에 담긴 대화 내용 또는 등장 인물의 목소리를 가급적 정확하게 전사함으로써 등장 인물의 움직임과 소리를 동시에 고려하여 해석적 작업을 하게 된다. C 프로그래밍 수업과 같이 학생이 어떻게 아이디어를 찾아서 발전시켜 나가는지를 파악하는 것은 학생과의 대화 내용, 얼굴 표정, C 프로그래밍 코드의 전개, 대화의 톤 등을 종합적으로 고려함으로써 가능하다 [2][7][16][17].

연구자는 비디오 화면과 동영상 캡처 소프트웨어 (예, 킴타시아) 자료를 같이 동기화해서 리뷰하면서 코딩과 해석 작업을 하게 되는데 종종 명확하지 못한 데이터와 해석하기 어려운 부분을 발견하게 된다. 학생이 작성했던 노트 자료를 참고하기도 하지만 해석 작업이 어려울 수 있다. 연구자가 비디오 촬영 중에 놓쳤던 질문을 발견하게 되고 특히 비디오를 통해 연구자가 잘못 해석한 부분이 있는지를 점검할 필요성이 생긴다. 이런 이유로 학생과 연구자가 함께 비디오를 보면서 기억을 더듬어 가며 연구자가 후속 면담을 실시하게 된다[11][12].

3.2 연구참여자 선정과 자료 수집

연구자가 운영하고 있는 컴퓨터 학원에 다니는 고등학생으로부터 같은 학교에 다니는 컴퓨터 프로그래밍의 경험을 가진 'K' 학생을 추천받았다. K에게 연구의 목적을 충분히 설명하고 면담을 요청하였고, 학생이 흔쾌히 면담과 그 내용의 녹음을 허락해주었다.

이 연구에서 자료 수집은 아래의 그림과 같은 순서로 면담이 진행되었다. K에게 연구의 목적을 충분히 설명하고 면담을 요청하였고, 학생이 흔쾌히 면담과 그 내용의 녹음을 허락해주었다.



<그림 1> 자료 수집 절차

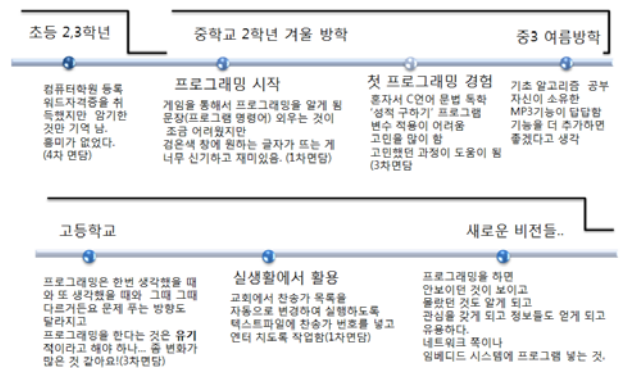
연구자는 2012년 6월에 연구참여자 K가 하노이 탑(Tower of Hanoi) 문제를 프로그래밍 하는 과정을 비디오로 촬영하였다. K가 평소에 프로그래밍을 하던 방식대로 비디오 촬영으로 인한 방해 를 최소화하여 그가 집중하여 프로그래밍을 할 수 있도록 비디오 촬영을 진행하였다. 중간에 혼자 생각하고 고민하는 부분에서는 10분씩 비디오를 멈추고 연구 참여자가 프로그래밍에만 전념할 수 있도록 하였다. 컴퓨터 프로그래밍 단계마다 연구자가 간단한 질문을 던졌고 연구 참여자는 자신이 해결해야 될 문제에 집중하면서 대답을 하는 형태로 자료 수집이 이루어졌다. 컴타시아 (Comtasia) 소프트웨어를 사용하여 K가 프로그래밍 하는 코딩 화면을 동영상 방식으로 녹화하였으며 연구자가 간간히 노트에 작성한 내용을 디지털 카메라로 찍어서 스캔하였다. 또한 그가 문제해결을 하는데 사용되는 아이디어를 손으로 쓰면서 유사 코드로 작성하는 손코딩 자료도 사진기로 촬영하여 디지털 이미지로 저장하였다[12].

3.3 연구참여자 소개

초등학교 2, 3학년 때는 컴퓨터자격증을 따기 위해 컴퓨터 학원을 조금 다녔지만 자격증을 딴 후 별로 흥미가 없어서 다니다가 그만 두었다. 그는 중학교 2학년 때 'GTA산안드레아스'라는 게임을 하면서 서버에 자신만의 모드를 구축하기 위해서 C 프로그래밍이 필요하다는 것을 주변사람들을 통해 듣게 되었다. 이것을 계기로 중학교 2학년 겨울방학부터 학원에 등록하여 C 언어를 배우기 시작했다. 처음 배울 때는 명령어 외는 것이 조금 어려웠지만 검은색 창(모니터에 나오는 결과화면)에 원하는 값이 나오는 것이 신기하였고 재미를 느끼게 되었다. 학원에서는 학생 혼자 스스로 공부하는 방식을 권장하여 C 언어 책을 혼자서 읽고 C 프로그래밍을 익혀 나갔게 되었지만 그다지 어렵다고 생각하지 않았다. 한 달 정도 기본 문법을 익히고 나서 학원선생님께서 내 주신 성적프로그램에 관한 문제를 프로그래밍하는 과정에서 고민을 많이 한 끝에 결국은 선생님의 도움으로 변수 처리에 대한 부분을 풀게 되었다.

프로그래밍을 하고 나서 처음에 고민하면서 풀

었던 과정들이 나중에는 도움이 되었음을 느꼈고 그러한 실패의 경험은 그가 다른 문제에 부딪혔을 때도 포기보다는 좀 더 도전해보게 하는 좋은 자양분이 되었다. 연구참여자 K는 처음 해 보았던 C 프로그래밍에 대한 경험을 아주 생생하게 기억하고 있었고 그의 목소리에는 여전히 풀지 못했던 그때에 대한 아쉬움이 묻어 나왔다.



<그림 2> 연구참여자 K의 프로그래밍 경험에 관한 시간 흐름표

그 이후에 두 달 정도 다니다가 다시 중3 여름방학 때 기초 알고리즘을 배웠다. 그때 배운 간단한 기초 알고리즘 배우고 나서 본인이 가지고 있던 MP3 기능이 너무 한정되어 있어서 좀 더 프로그램을 넣어 확장해 보고 싶은 생각들을 했으며 그런 생각들이 그때는 너무 재미있다고 회고했다. 고등학교 입학 후 학교에서 정보올림피아드 대회 참가를 위해 선발되어서 다시 프로그램을 하게 되었고 처음 출전한 전남대회에서 동상을 탔다. 그것을 계기로 C언어로 프로그래밍 공부를 꾸준히 되었으며 고등학교 2학년 때 전남 올림피아드 대회에서 은상을 받았다. 그 후에는 학교내 기능반 수업 일정으로 주말에만 정보올림피아드 준비를 위해 프로그래밍 학원에 다니고 있다.

이렇듯 컴퓨터 프로그래밍 경험을 통해 그는 학교 공부를 좀 더 계획적으로 할 수 있게 되었고 수학시간에도 함정이 많은 문제를 만날 때 여러 가지 방향으로 생각하고 함정을 미리 찾고 들어가기 때문에 도움을 많이 받곤 하였다. 또한 프로그래밍을 통해 평소에 주변에 관심이 없던 새로운 분야에 대해서도 관심을 갖게 되었으며, 정

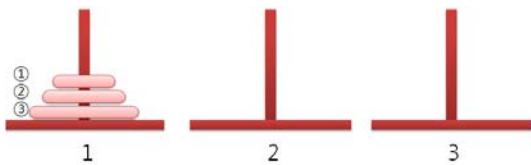
보를 찾아보게 되어서 안 보이던 것도 보이게 되고 몰랐던 것도 알게 되어 K 자신에게는 무엇보다도 유용한 공부거리가 되었다.

4. 하노이 탑 프로그래밍 과정에 대한 질적 사례

4.1 하노이 탑 문제 분석 및 설계

문제 분석은 자신의 생각을 탐색하는 과정으로 문제를 정의하고 타당성, 능률성, 경제성 등을 개괄적으로 검토하는 것으로 프로그래밍의 첫 번째 단계로 문제를 읽고 이해하는 것이다. 문제를 이해하는 것은 과학적 사고와 비판적 사고의 가장 기본이 되는 것으로 문제의 핵심을 파악하고 전체의 구조와 필요한 요소, 형태를 하나씩 분석해보고 자신이 알고 있는 지식안에서 연결고리를 찾아보고 다양한 각도에서 생각을 해 보는 것이다.

하노이 탑은 동판에 세 개의 막대가 있고, 크기가 다른 3개의 원판이 한 막대에 꽂혀 있을 때 원판을 다른 막대로 모두 옮기는 놀이이다. 한 번에 한 개의 원판만을 옮겨야 하며 크기가 큰 원판은 반드시 크기가 작은 원판 아래쪽에 있어야 한다. <그림 3>처럼 왼쪽의 첫 번째 막대에 있는 원판이 세 번째 막대의 원판으로 모두 이동이 되어야 한다.



<그림 3> 하노이 탑

K는 통상 프로그래밍 과제에 대한 지문을 많이 읽어보고 차근 차근 노트를 준비를 해서 노트에 다 문제를 적기도 하고 천천히 수용도도 그리고 표도 그리고 그림을 그리면서 쉽게 이해할 수 있도록 그리면서 답을 찾아 가는 것이 자신만의 방식이라고 한다. 이렇게 문제를 적고, 수용도나 표를 그리는 행동 자체는 프로그래밍을 설계하는 것으로 자신의 생각을 탐색한 후에 구체화시키는 작업으로 문제 해결력에서 중요한 방법인 브레인

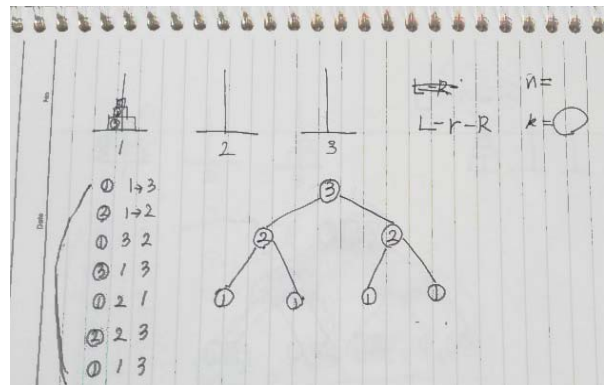
스토밍과 같은 열린 사고체제를 이끌어내는 요소이다.

K에 대한 면담 자료를 바탕으로 작성된 아래의 표를 살펴보면 그가 프로그래밍을 하면서 문제를 분석하고 설계하는 방법을 통해 다양한 과학적 사고가 발현되는 것을 알 수 있다.

| 면담 코딩 요약 | 과학적 사고 |
|---|--|
| <ul style="list-style-type: none"> ◦ 그 문제가 바로 이해되는 것이 아니니까, 계속해서 많이 문제를 읽어보고 (1차면담) | <ul style="list-style-type: none"> ◦ 문제 이해력 ◦ 절차적 사고 |
| <ul style="list-style-type: none"> ◦ 어, 저는 따로 우선은 차근차근 노트를 준비해서 거기에다 문제를 쓰고 ◦ 만약에 수용도가 필요하면 수용도를 그리고 표가 필요하면 표도 그리고 그림을 그리면서 좀더 쉽게 이해할 수 있도록 그리면서 거기에서 제가 답을 찾는 거 같아요. 그리고 저는 한번 손코딩을 해봐요. (2차면담) | <ul style="list-style-type: none"> ◦ 문제 이해력 ◦ 절차적 사고 ◦ 분석적 사고 |
| <ul style="list-style-type: none"> ◦ 문제를 풀기 전에 보다가 숫자를 쓴 순간 순간 마다 패턴을 한번 분석하는 경향이 있는 거 같아요! 순서대로 쓰다보면 보이는 패턴들이 있고 패턴이 보이는 순간 '이렇게 쓰면 되겠다' 라는 생각이 연결이 돼요! (4차 후속 면담) | <ul style="list-style-type: none"> ◦ 분석적 사고 ◦ 논리적 추론 |

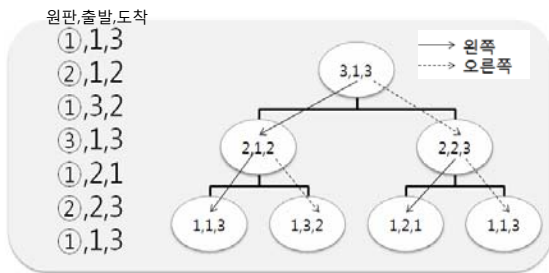
<표 1> 면담 코딩의 요약

비록 그는 고등학교 1학년때 재귀 함수를 공부하면서 하노이 탑 문제를 접해 보았지만 그 당시에는 문제를 해결할 수 없었다. 그러나 이번 연구를 위해 제시된 하노이 탑 과제를 접했을 때는 맨 처음 트리가 생각난다고 했다. 하노이 탑의 원판을 옮기는 순서를 써보니 중위방식의 이진 트리가 생각나서 종이에 그림을 그렸다.



<그림 4> 수용도 그리기

<그림 4>는 K가 문제를 읽고 하노이 탑 원판을 옮기는 순서를 종이에 적은 뒤 이진 트리 표현한 것이다. 숫자 ①②③ 은 원판을 나타내며 ① 1→3 은 첫 번째 원판을 1번 막대에서 3번 막대로 옮긴다는 의미이다. 숫자의 패턴을 보니 1213121 로 중위 방식이었고, 그 패턴에 따라 이진 트리를 그렸다. 기존에 자신이 알고 있던 지식을 활용하여 문제를 이해하고 분석하는 과정에서 논리적, 비판적사고가 나타나고 있음을 알 수 있다.



<그림 5> 이진 트리에서 패턴 찾기

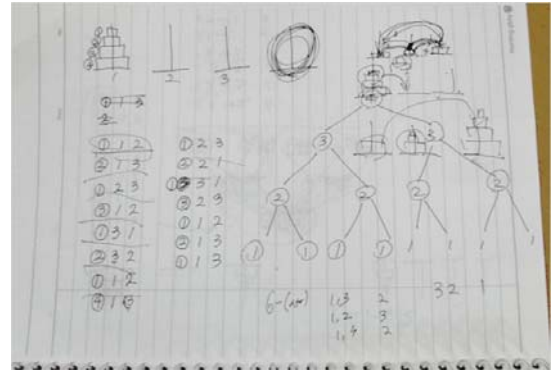
<그림 5>를 보면 이진트리 동그라미 숫자 옆에 막대 번호를 쓰고 패턴을 찾고 있다.

| 비디오 전사 자료(07:44 - 08:20) | 과학적 사고 |
|--|--------|
| K: 동그라미 부분은 잘 수 있는 부분이고 나머지 부분은 어떻게 잘 것인가? 보니까 제가 본건데 가운데 1은 왼쪽으로 갈 때는 똑같은 숫자가 가운데로 들어가고... 연구자: 규칙성을 찾는 건가요? K: 패턴... 오른쪽으로 갈때는 뒤에 있는 숫자는 똑같이 들어가고.. 그게 맞는지 보고 있어요! 그렇게 되는 거 같네요. 나머지 빈칸은 아아...찾아봐야죠! (계속 생각 중) | 논리적 사고 |
| | 비판적 사고 |

<표 2 > 패턴 찾기 비디오 전사자료

패턴을 살펴보니 이진트리에서 오른쪽으로 갈 때는 맨 뒤의 숫자가 똑같이 들어가고 왼쪽으로 갈때는 두 번째 숫자가 똑같이 들어간다. 왼쪽으로 갈 때 3번째 변수의 규칙이 (1,3 → 2), (1,2 → 3), (2,3 → 1)이고 오른쪽으로 갈 때 2번째 변수 (1,3 → 2), (1,2 → 3), (2,3 → 1)로 동일하다는 것을 찾았으며 각각의 변수의 합이 6이므로 6에서 위에서 전달되는 두 개의 변수를 빼주면 해결

된다는 것을 알게 되었다. 그래서 K는 자신이 찾아낸 패턴이 맞게 들어가는지 검증해본다.



<그림 6> 원판이 4개일 때 패턴 찾기

<그림 6>은 K가 자신의 이론을 검증하기 위해 하노이 탑의 원판이 4개일 경우를 가정하여 앞에서 했던 방법과 동일하게 원판의 이동 순서를 적고, 그 이동 순서에 따라 중위방식으로 이진 트리를 그렸다. 이러한 과정 속에서 문제를 계속적으로 반복하여 사고하는 재귀적 사고와 자신의 가설을 검증하는 비판적, 분석적, 논리적 사고가 나타나고 있음을 알 수 있다.

K는 프로그래밍을 할 때 주로 손으로 미리 코딩을 해보는데 프로그래밍 과정에서 어려웠거나 나중에 필요할 것 같은 내용은 기록을 해 놓는다 했다. 이러한 과정들이 알고리즘적이며 재귀적인 사고의 근간이 된다는 것을 유추할 수 있다.

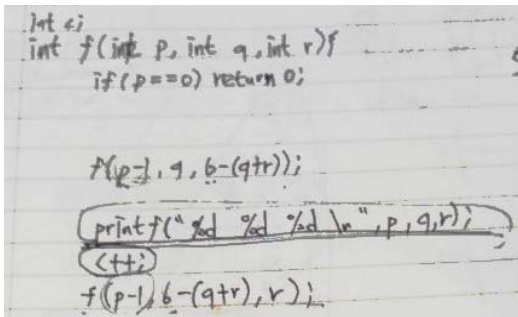
“연구자: 프로그램 다 짜고 나면 어려웠거나 어렵게 풀거나 나중에 필요하겠다 싶으면 손 코딩해 놓는다 했잖아요. 이 프로그램이 잘 짰는지 안 됐는지 어떻게 알아요? 마지막에 다 짜고 나서

K: 아... 굳이 테스트를...제가 할 수 있는 범위 안에서 해요! 그리고 맞다면 그렇게 하고.. 제 손으로 푼거하고 컴퓨터로 작업한 거하고 비교해보고 맞춰보고 그렇게 해서 맞았다고 판단할 수도 있고... 대부분 그렇게 하고 그 다음에 가끔 큰 숫자까지 돌려보겠다는 생각에 알고리즘 사이트가 있어요. 그래서 거기서 이제 문제들이 있는데

그 문제에 대한 소스를 거기다 넣고 입력을 해 놓으면 거기에 있는 예제파일들이 알아서 채점을 해서 점수가 나와요 그런게 있어요. (비디오 회상 면담)

4.2 문제 해결의 절차 및 구현

하노이 탑 문제를 읽고, 분석한 후 숫자의 패턴을 발견한 후 K는 종이에 C언어로 직접 코딩을 했다. K는 평소 프로그래밍을 할 때 손으로 직접 코딩을 해본 후에 실제 프로그래밍을 하였다.



<그림 7> 손코딩 자료

<그림 7>은 K가 C언어로 손 코딩한 자료로 메인에서 F(p, q, r)라는 함수를 부르고 왼쪽 트리 : f(p-1, q, 6-(q+r)), 오른쪽 트리: f(p-1, 6-(q+r), r)형태가 되었다. 여기서 P는 동그라미 숫자인 원판①을 나타내고, q는 출발 막대1 r은 도착 막대3을 나타낸다. 또한 중위 방식으로 출력하기 위해 중간에 printf를 넣었다.

프로그래밍의 부분인 main()의 일부분과 재귀 함수 중 첫 번째 변수인 탑의 이동 순서만을 중위방식으로 코딩을 하였다. 그리고 그 내용을 컴퓨터를 이용하여 직접 코딩, 실행 한 후 그 결과를 확인하였다.

“연구자: 설계단계에서 다 끝내기 못하는 경우도 있나요?”

K: 그...뭐냐면...손코딩을 하는게 완벽한 패턴이 보일 때도 있겠지만 중간에 패턴이 보이고 그걸 실험하면서 손코딩하고 실험하고

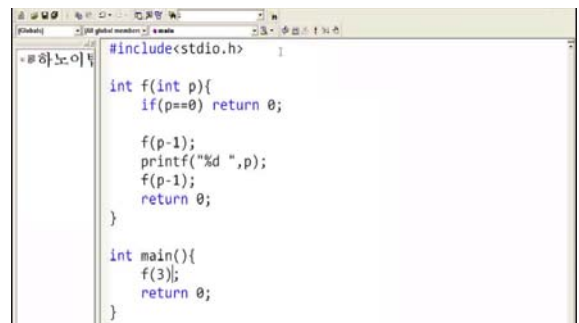
연구자: 실험한다는 거는 컴퓨터로 테스트를

한다는 건가요?

K: 네. 그런식으로 하면서 중간 중간 맞는지 확인하면서 그런식으로 프로그램이 가끔 부분 부분이 있어요. 정렬, 처리, 그런 식으로 부분 부분이 있는데 그런 부분 부분을 설계, 손코딩 하고 설계, 손코딩, 직접 실험.. 계속 그렇게 (비디오 회상 면담)

위 비디오 회상 면담 자료를 보면 손으로 코딩할 때 프로그래밍의 전체를 작성해 나가는 것이 아니라 프로그램의 부분 부분을 코딩하면서 실험하고 다시 코딩하는 과정이 반복되는 것을 알 수 있다. 이러한 과정속에서 논리적이며 비판적인 사고과정이 일어나고 있음을 알 수 있다.

<그림 8>은 K가 실제 프로그래밍한 화면인데 메인에서 3개의 탑을 처리하기 위해 재귀 함수 F(3)를 호출한다.



<그림 8> 코딩 캡타시아 캡처 화면

F함수는 재귀함수이며 원판의 숫자들이 중위 방식으로 이동하여 출력하기위해서 재귀함수 F(p-1)→ printf(p) → F(p-1) 순서대로 코딩을 한 후에 결과를 출력한 것이다.



<그림 9> 결과확인: 캡타시아 캡처 화면

컴퓨터 프로그래밍에서 자신의 생각을 실험할 때 중요한 부분 중에 하나는 프로그램을 실행하는 도중에 예상치 않은 오류가 발생하고 그 오류를 수정하는 과정을 경험한다는 것이다. 아주 간

단한 컴파일 오류에서부터 링크오류, 또는 설계과정으로 다시 피드백을 해야 되는 구조적 오류까지 다양한 경험을 갖게 된다. 이러한 다양한 오류를 디버깅(debugging; 오류수정) 과정에서 프로그래밍의 각 단계별로 차근차근 따라가면서 오류의 원인을 찾아내고 특정 위치에 변수를 놓거나 그 값의 출력을 통해 결과를 확인함으로써 절차를 계속해서 따라가고 평가하는 사고를 반복적으로 사용하게 된다.



<그림 10> 여러 수정 캡타시아 캡처 화면

4.3 “자신만의 생각을 실험하다”

K의 사례에서 두드러지게 나타나는 부분은 자신만의 생각을 실험하는 단계이다. 그는 하노이탑 문제를 컴퓨터로 해결할 때 프로그래밍이 어떻게 돌아가는지 차근차근 적용해 보고 어느 알고리즘이랑 연관이 되는지 한번 더 생각해 본 후에 자신만의 방식으로 실험을 해본다고 했다.

프로그래밍을 구현하기 위해 K는 주어진 문제를 파악하고 노트에 손으로 프로그래밍을 써내려갔다. 그러면서 중간 중간에 생각이 맞는지 확인하면서 손으로 프로그래밍을 써 내려갔다. 또한 처음에 생각했던 답이 나오지 않는다 할지라도 그 실행을 통해 새로운 방식을 다시 탐색하거나 원하는 부분을 먼저 실행 한 후 그 결과가 적절한지를 확인하고 다시 생각을 구체화 하는 과정을 거쳤다. 이렇듯 그는 프로그래밍을 통하여 주어진 문제를 스스로 이해하고 해결하기 위해 노력함으로써 탐구학습에서 나타나는 과학적 사고에 근접한 양상을 보여주었다.

면담 요약

여러 가지 방식으로 프로그램을 작성해 보면서 배웠던 기억들을 생각해 보고 실행을 해봅니다. (1차면담)

코드를 몇 번 돌고 난 후에 정지하는 코드 값을 것을 제거 원하는 부분에다 놓고 정지하는 부분마다 프린트를 해요! 그러니까 출력을 한번 해봐요 어느 변수가 어떻게 돌아가는지 하나씩 하나씩 차근 차근 하면서 이 부분이 틀렸다면 이 부분이 왜 틀렸는지 다시 생각해 보고 그것에 다시 또 방법을 세워서 오류를 수정해 나가는... (2차 면담)

K: 손코딩을 하는게 완벽한 패턴이 보일 때도 있겠지만 중간에 패턴이 보이고 그걸 실험하면서 손코딩하고 실험하고 그런식으로 하면서 중간 중간 맞는지 확인하면서 그런 식으로 프로그램의 부분 부분이 있어요 정렬, 처리, 그런 식으로 부분 부분을 설계, 손코딩하고 설계, 손코딩, 직접 실험.. 계속 그렇게 (4차 후속 면담)

<표 3> 자신만의 생각을 실험하기에 관한 면담

컴퓨터 프로그래밍은 보다 정확한 정보를 체계적인 방법으로 구성하고 그 정보에 논리적인 조작을 하도록 환경을 마련해 줌으로써 자연스럽게 절차적 사고와 논리적 추론을 하도록 돕는다. 그리고 전체의 문제를 해결 가능한 작은 부분으로 나누어 구체적인 해결 방안을 찾아가도록 하는 분할적 사고를 사용하게 한다.

프로그래밍을 하고 나서 실행 시켰을 때 오류가 종종 발생하는데 K는 오류가 발생할 때 코드를 몇 번 실행시켜보고 정지하는 코드를 원하는 부분에 놓고 정지하는 부분마다 프린트를 해서 어느 변수가 어떻게 돌아가는지 하나씩 하나씩 차근차근 하면서 이 부분이 틀렸다면 왜 틀렸는지 다시 생각해보고 그것에 따른 방법을 생각해서 다시 실행해 보았다. 그는 자신의 생각을 실험할 때 예기치 못한 오류가 발생하여 오류를 찾아 수정하는 작업을 반복적으로 실행하면서 결과에 따른 과정을 다시 한번 살펴보고 그런 일련의 절차를 따라 자신의 작업을 스스로 집어보는 패턴을 보여주었다.

4.4 실생활에서의 적용

C 프로그래밍 하기 전에는 K는 계획도 없이 막연한 생각으로 공부를 해서 실수를 많이 했지만 프로그래밍을 한 후에는 계획을 세우면서 하

니까 공부도 잘되고 실수를 적게 하게 되었으며 공부에 자신감을 갖게 되었다. 특히 수학기간에는 수업할 내용에 대해 한 번 생각해보고 머릿속에서 계속 생각을 하면서 선생님께서 다음에 무엇을 말할지를 계속 따라가면서 수업을 하게 되어서 수업도 즐겁고 잘할 수 있게 되었다. 또한 절차적인 사고들과 추론적 사고를 하면서 그가 함정이 많은 수학문제들을 어떻게 대하는지 보면 과학적 사고력과 직결된 응용력이 발휘되는 사례를 부각시키게 된다.

“진짜 프로그래밍을 하면 안 보인 것도 보인다고 해야 하나, 내가 몰랐던 것도 알게 되고 좀 더 찾아보게 되고 관심도 갖게 되니까...” 또한 K는 관심있는 기술을 적용하는 방법들에 대해 책을 사서 보거나 정보를 구체적으로 찾아보면서 컴퓨터 프로그래밍 경험을 통해 자신의 역량을 확장시키게 되었다. 그만의 방식에 맞는 진짜 프로그래밍을 해 보는 과정에서 문제를 해결하는 방식이 막히거나 그의 생각과 다른 결과들이 나올 때 K 나름대로 풀어나갈 수 있는 방법도 찾아보고, 어려가지 다양한 문제를 해결했던 경험을 갖게 되어 자신감을 얻게 되었다. 자연스럽게 그는 어떠한 일을 하더라도 포기하기보다는 그런 상황을 해결하려는 노력을 많이 하게 되었다.

5. 결론 및 제언

본 연구에서는 주어진 과제를 컴퓨터 프로그래밍하는 과정에서 나타나는 연구 대상 학생의 사고의 패턴을 분석하고자 했다. 하노이 탑 문제를 이해하고 분석하는 과정에서 기존의 지식을 활용하면서 재귀적이며 알고리즘적인 사고를 하였고, 프로그래밍 설계를 하는 과정에서는 논리적이고 비판적이며 분석적인 사고를 동반하였으며 프로그래밍을 실제 구현하는 과정에서는 재귀적, 비판적, 논리적인 사고가 나타남을 알 수 있었다. 문제를 분석하는 과정에서 주어진 문제를 분석하고 아이디어를 탐색하고 설계하고 구현하는 이 모든 과정에서 재귀적, 논리적, 비판적 사고가 고루 나타나고 있음을 알 수 있었다.

본 연구에서는 단일 학생의 사례를 바탕으로 면담과 비디오 촬영, 비디오 회상 면담 자료 등

다양한 자료를 바탕으로 해석하였지만 한 명의 사례가 가지고 있는 제한점이 있으므로 좀 더 많은 사례를 수집하여 컴퓨터 프로그래밍과 관련된 다양한 정보과학적 사고의 발현이 어떻게 이루어지고 있는지에 대한 연구가 더 필요할 것이다 [2][7][15]. 이러한 연구를 통해 실제 현장에서 교육을 진행하는 교사나 프로그래밍을 공부하는 학생들에게 보다 창의적이고 과학적인 교육방법을 제시 할 수 있으리라 기대해본다.

5.1 해석적 논의

본 연구에서 우리는 K가 컴퓨터 프로그래밍 경험을 통해 하노이 탑 문제를 분석하고 처리하는 과정에서 자신만의 방식으로 생각하며 프로그래밍 코딩을 하는 고유한 패턴을 포착할 수 있었다. 그는 컴퓨터 프로그래밍을 할 때 ‘처음에 문제를 천천히, 차근차근 읽어보고 직접 노트에 쓰기도 하고, 그 문제를 분석해서 패턴들을 찾아내고 여러 가지 방식으로 생각을 구체화하여 알고리즘을 적용하고 컴퓨터로 구현해 나가는 것’ 방식을 드러내 보였다. 문제해결에 관한 단서가 되는 아이디어를 혼자서 생각해보면서 자신의 힘으로 문제를 차근히 대하면서 손코딩 해 가는 ‘자기만의 생각을 실험하는 방식’에 충실한 K는 자신만의 아이디어로 컴퓨터 조립과 프로그래밍 작업을 통하여 새로운 것을 만들어가는 김상우 사례와 관련성이 매우 높게 나타나고 있다[2].

컴퓨터 프로그래밍 경험을 통해 나타난 이러한 사고 과정은 문제를 이해하는 것뿐만 아니라 처리 과정 속에 절차적이고 추론적인 사고들이 계속해서 연관되어 나타남을 알 수 있었다. K는 이러한 프로그래밍 작업에 수반되는 논리적이고 절차적인 능력이 수학과 같이 다른 수업에서도 공부에 도움이 된다는 점을 인식하고 있었다. 또한 그는 비디오 화면을 통하여 상당한 집중력을 발휘하면서 손 코딩과 프로그래밍 코딩을 하였고 디버깅을 하는 모습을 보여주었다.

K에게 다시 한번 생각하게 하고, 여러 가지로 생각 할 수 있도록 그 맥락을 이어주는 것은 컴퓨터 프로그래밍 경험을 통해서 얻었던 ‘생각하는 것’에 대한 즐거움이었다. 그는 그만의 방식으로

컴퓨터 프로그래밍을 통해 문제를 푸는 과정에서 문제가 막힐 때마다 계속해서 생각하는 과정을 통해 아이디어를 찾고 해결해나가곤 하였다. 그는 문제에 대한 패턴 분석을 할 때도 나름대로 규칙을 세워 생각하고 찾아가는 방법에 재미를 느꼈으며 시간이 지나서 다시 문제를 볼 때에도 나중에 새로운 아이디어가 떠오르는 것을 즐겨하였다. K는 이러한 컴퓨터 프로그래밍의 경험을 통해 자신이 발전해 가고 있다고 생각하고 있었으며 문제를 풀 때의 어렵고 힘든 과정을 마치 산을 정복했을 때의 성취감을 맛보는 것과 같은 즐거움을 느끼는 과정으로 여기고 있었다.

심층면담과 비디오 촬영을 통하여 나타나는 데이터를 해석한 결과 K가 C 프로그래밍 작업을 통하여 보여주는 자기만의 방식, 즉 문제를 천천히 다루고 노트에 적어보기도 하면서 도표 또는 그림으로 차근차근 아이디어를 모색하는 그만의 방식이 고유하게 나타남을 볼 수 있었다[2]. 그런 과정에서 자신의 아이디어를 더 발전시키고 다른 아이디어와 비교하면서 다른 각도에서 모색하는 과정에서 C 프로그래밍 코딩을 하고 결과를 즉시 확인하는 일련의 작업이 마치 과학자가 실험하듯이 자신의 생각을 컴퓨터로 구현시키는 작업이 어렵기도 하였지만 C 프로그래밍은 자신의 내적인 지적 욕구를 충족시킬 수 있는 하나의 매개체로 작용하고 있었다.

본 연구를 통해 나타난 컴퓨터 프로그래밍 경험과 과학적 사고 사이의 관련성을 논의한 내용은 다음과 같다. 첫째, 문제를 분석하고 설계하는 과정에서 끊임없이 생각들을 끄집어내고 문제를 읽고 분석하면서 자유롭게 다양한 생각들을 문제에 적용해 가는 과정을 겪는다. 이러한 경험은 아이디어 탐색과정과 매우 유사하며 문제 이해력을 바탕으로 분석적 사고와 브레인스토밍과 같은 다양한 사고의 발현과 통찰을 갖게 한다. 둘째, 자신의 생각을 실제 프로그래밍을 통해 실험해봄으로서 정확한 정보를 체계적인 방법으로 구성하고 그 정보에 논리적인 조작을 함으로써 자연스럽게 과학적 사고의 기본이 되는 절차적 사고와 논리적 추론, 비판적 사고 등이 관련되어 나타나게 된다. 셋째, 컴퓨터 프로그래밍 경험 통해 과학적 사고의 주요 기능 중 하나인 사고를 확장하고 적

용할 수 있다는 것이다. K의 경험에서 나타나듯이 현재 자신의 삶 속에서 프로그래밍 경험에서 얻은 계획적이고 절차적이며 논리적인 사고들을 사용하고 있으며 실제로 문제해결이 필요한 경우 K 스스로 그 문제를 해결하기 위해 도전하고 노력한다는 점이 두드러지게 나타나고 있다.

5.2 시사점과 제언

본 연구의 해석적 결과가 컴퓨터 프로그래밍과 관련된 컴퓨터교육과정에 시사하는 바는 다음과 같다. 컴퓨터 프로그래밍을 한다는 것은 새로운 문제를 절차적이고 논리적으로 코딩하여 해결책과 다른 결과가 나왔을 때 오류를 수정-보완해야 하는 순환적인 작업 구조와 깊이 관련되어 있다. 무엇인가를 처음부터 생각해야 하고 그것을 해결해야 하는 방법을 생각해 내야하며 스스로 방향을 정해서 결정하고 실행해서 결과를 만들어 내야하는 이 모든 단계가 프로그래밍 과정 가운데 있다. 컴퓨터 프로그래밍을 통한 문제해결의 장점은 학생으로 하여금 그 문제의 핵심을 파악하고 그 문제의 전체의 구조와 필요한 자료의 요소와 형태를 세세하게 분석하게 함으로써 학생 자신이 가지고 있는 총체적 지식의 연결고리를 찾아보게 하고 그 문제를 여러 가지 각도에서 생각해 보게 하는 것이다[5][6].

그러므로 컴퓨터 프로그래밍을 경험하는 학생들은 보다 정확한 정보를 체계적인 방법으로 구성하고 그 정보에 논리적인 조작을 하도록 환경을 마련해 줌으로써 자연스럽게 과학적 사고력을 신장시킬 수 있다. 정보과학적 사고를 이루는 절차적 사고와 논리적 추론을 활성화 시킴으로써 복잡한 융합형 문제를 해결 가능한 작은 부분으로 나누어 구체적인 해결방안을 찾아낼 수 있다. 또한 컴퓨터 프로그래밍을 통하여 자신의 생각을 실험하는 과정에서 프로그램 작동에 예상치 않은 오류가 발생하면 그 오류를 수정하는 과정을 경험하게 된다. 이러한 과정에서 프로그래밍의 각 단계를 차근차근 따라가면서 오류의 원인을 찾아내고 특정 위치에 변수를 놓고 그 값의 출력력을 통해 결과를 확인함으로써 절차를 계속해서 따라가고 평가하는 사고를 반복적으로 사용하게 된다.

본 연구가 던지는 향후 방향은 다음과 같다. 최근 놀이와 교육을 접목한 LOGO언어, 로봇을 활용한 프로그래밍 교육 및 영재교육에 관심이 높아지고 있는데 이러한 놀이와 로봇 및 프로그래밍 과정을 통해 나타나는 과학적 사고와 창의적 활동이 발현되는 과정에 대하여 본 연구에서 진행한 방법론을 적용할 수 있다[9][11][12]. 한 명을 대상으로 심층 면담기법과 비디오 촬영 및 비디오 후속 면담 등을 사용하여 주제에 접근하였지만 2-3명의 연구참여자를 대상으로 연구를 진행함으로써 사례간 비교고찰을 하는 것이 바람직하다. 또한 하노이 탑 이외의 과제에 대한 컴퓨터 프로그래밍 경험 속에서 다양한 과학적 사고의 발현이 어떻게 이루어지는지에 대하여 초상화법과 같은 심층연구가 더 필요할 것으로 생각된다 [2][16][20].

참 고 문 헌

- [1] 교육인적자원부(2005). 초·중등학교 정보통신 기술 교육 운영지침.
- [2] 전영국, 윤지현(2011). ‘나만의 것’을 추구하는 컴퓨터 교육과 재학생의 소프트웨어 개발 경험에 관한 질적 사례연구: 컴퓨팅에 관한 심층 동기와 개인 성장 및 비전을 중심으로, 교육인류학회 논문지. 14(1), 43-76.
- [3] 송상호(2012). 로봇활용수업의 에듀테인먼트적 특성 탐색. 교육정보미디어연구, 18(4), 389-416.
- [4] 김수환, 한선관, 김현철(2010). 프로그래밍 과정에서 나타나는 초보학습자들의 행동 및 사고과정 분석. 한국컴퓨터교육학회 논문지, 14(1), 13-21.
- [5] 문외식(2005). 초등학생의 논리적 사고력 및 문제 해결 능력 향상을 위한 컴퓨터 프로그래밍 교육과정 모델제안. 정보교육학회 논문지, 9(4), 595-605.
- [6] 한재협(2010). 논리력과 문제해결력 신장을 위한 EPL기반 교육전략. 석사학위논문, 경인교육대학교.
- [7] 홍용희, 노경주, 심종희 역(2000). 질적 사례 연구. 서울: 창지사.
- [8] CS4FN. <http://www.cs4fn.org>
- [9] 전영국, 김진영, 김정섭, 조인성, 허영현(2006). 머리가 좋아지는 Logo 프로그래밍. 교우사.
- [10] 김영천(2006). 질적 연구 방법론 I. 문음사, 399.
- [11] 윤지현(2004). 비디오 분석기법을 이용한 정보과학 영재 학생의 질적 사례 연구. 순천대학교 교육대학원 석사 학위 논문.
- [12] 장정숙 (2012). 컴퓨터프로그래밍 경험과 관련된 과학적 사고에 관한 질적 사례 연구. 순천대학교 교육대학원 논문.
- [13] Frazier, R. (1996). Ways of working, ways of being: A study of four children in a setting for learning science. Unpublished doctoral dissertation, University of Illinois, Urbana-Champaign.
- [14] Rosenstein, B. (2002). Video use in social science research and program evaluation. International Journal of Qualitative Methods, 1(3). Article 2. Retrieved DATE from <http://www.ualberta.ca/~ijqm>
- [15] John W. Creswell(2007). 조홍식·정선옥·김진숙·권지성 공역 (2010), 『질적 연구방법론(다섯가지 접근)』, 학지사
- [16] Witz, K. (2006). The participant as ally and essentialist portraiture. Qualitative Inquiry, 12, 246-268.
- [17] Witz, K., Goodwin, D., Hart, R. S., & Thomas, S. (2001). An essentialist methodology in education-related research using in-depth interviews. Journal of Curriculum Studies, 33(2), 195-227.
- [18] 김수환(2011). Computational Literacy 교육을 통한 문제 해결력 향상에 관한 연구, 고려대학교 박사학위 논문.
- [19] 전영국, 명성원, 심혜영(2006). 중등 컴퓨터 예비교사의 진학 동기, 교과교육학 지식, 전공 공부 및 비전 형성. 컴퓨터교육학회논문지, 9(3), 13-28.
- [20] 주수희(2006). 프로그래밍 교육을 통한 정보과학 영재의 질적 사례 연구. 순천대학교 교육대학원 석사학위 논문.
- [21] Dunican, E. (2005). A Framework for Evaluating Qualitative Research Methods in Computer Programming Education In P.

Romero, J. Good, E. Acosta Chaparro & S. Bryant (Eds). Proc. PPIG 17, pp-255-267.



장 정 숙

1994 군산대학교
컴퓨터과학과(학사)
2012 순천대학교 컴퓨터교육과
(교육학석사)

2012~현재 여수 안산중학교 정보컴퓨터 교사
관심분야: 컴퓨터교육, 프로그래밍 교육
E-Mail: ncajs3@nate.com



전 영 국

1986 수원대학교 수학과
(이학사)
1990 시카고주립대 수학과
(이학석사)

1995 Univ. of Illinois Urbana-Champaign
(컴퓨터교육 박사)
1996~현재 순천대학교 컴퓨터교육과 교수
관심분야: 지능형교수시스템, WBI, 질적 연구
E-Mail: ycjun@sunchon.ac.kr



윤 지 현

1998 순천대학교
정보통신공학과(공학사)
2005 순천대학교
컴퓨터교육과(교육학석사)

2010~현재 순천대학교 과학정보융합학과
박사과정
관심분야: 컴퓨터교육, 질적 연구-비디오 분석,
심층면담
E-Mail: skysak@nate.com