

짝프로그래밍 협업도구를 활용한 SW교육프로그램

김용옥* · 전석주**

서울서래초등학교* · 서울교육대학교**

요약

4차 산업혁명에 따라 빠르게 변화되는 사회에서 미래 인재는 지식의 양이 아닌 질적으로 수준 높은 지식을 활용하고 구성하는 능력이 요구되고 있다. 2015 개정 교육과정의 출범에 따라 역량 중심으로 교육과정이 개편되었으며 소프트웨어 교육에서는 '정보문화소양', '컴퓨팅 사고력', '협력적 문제 해결력'의 역량을 추구하고 있다. 본 연구에서는 미래인재의 역량인 협력적 문제해결력을 함양하기 위해서는 교육 현장에서 교사가 손쉽게 활용 가능한 협업도구와 이를 활용한 SW교육 프로그램을 개발한다. 이러한 협업도구를 활용한 SW교육프로그램은 프로그래밍 언어와 문법에 대한 지식, 프로그래밍 기법에 대해 배우는 것보다 짝프로그래밍을 통해 컴퓨팅 사고력과 협력적 문제해결력을 함양할 수 있다. 총 16차시로 구성된 수업에서 협업도구를 활용한 짝프로그래밍을 통해 수업이 참가한 학생들의 협력적 문제해결력과 코딩 이해도를 향상시킴을 보여준다.

주요어 : 소프트웨어 교육, 짝프로그래밍, 컴퓨팅사고력, 협업 도구, 협력적 문제해결력

SW Education Program using Pair Programming Collaboration Tools

Yong-Ok Kim* · Seok-Ju Chun**

Seoul Seorae Elementary School* · Seoul National University of Education**

ABSTRACT

In a rapidly changing society with the Fourth Industrial Revolution, future students should have the skills to utilize and organize high-quality knowledge, not the amount of knowledge. With the launch of the revised curriculum in 2015, the curriculum has been reorganized based on competency and the software education has been seeking capabilities such as 'Cultural computing knowledge', 'Computational thinking', and 'Collaborative problem solving skills'. Therefore, practical collaboration tools and education programs that can be used in the field of education are developed based on Pair Programming, which is a specific collaborative learning strategy to develop cooperative problem solving skill. The educational program using this collaboration tool was developed with a focus on developing computational thinking and collaborative problem solving skills through Pair Programming rather than focusing on learning grammar of programming language and programming techniques. In a educational program, students will be able to use collaborative tools for pair programming and foster collaborative problem-solving skills.

Keywords : SW education, Pair programming, Computational Thinking, Collaboration tools, Collaborative problem solving skills

교신 저자 : 전석주(서울교육대학교 컴퓨터교육학과)

논문투고 : 2019-06-15

논문심사 : 2019-06-18

심사완료 : 2019-08-26

1. 서론

4차 산업혁명에 따라 빠르게 변화되는 사회에 따라 학교 교육의 방향성도 지식과 기능의 측면에 중점을 두었던 바를 지식기반사회를 살아가는데 필요한 핵심역량에 두게 되었다. OECD는 이러한 역량을 21세기 사회에서 성공적인 삶을 살기위해 필요한 능력으로 ‘자율적으로 행동하기’, ‘이질적인 집단에서 상호작용하기’, ‘도구를 상호적으로 사용하기’를 선정하였다[11].

2019년부터 초등 실과 과목에서 운영되는 소프트웨어 교육에서 추구하고 있는 역량은 크게 ‘정보문화소양’, ‘컴퓨팅 사고력’, ‘협력적 문제 해결력’으로 나뉜다. 그 중 협력적 문제해결력은 ‘협력적 컴퓨팅 사고력’, ‘디지털 의사소통능력’, ‘공유와 협업능력’을 포함한다[12]. 그 중요성에 따라 현재까지 교육 현장에 적용할 수 있는 다양한 컴퓨팅 사고력을 증진시킬 수 있는 교수학습방법 및 자료를 연구가 활발히 진행되고 있다. 그러나 초등학생을 대상으로 협력적 문제 해결을 효과적으로 함양하기 위한 교육 방법은 교사로 하여금 수업 현장에서 보다 추상적으로 느껴질 수 있으며 교사를 위한 구체적인 협력적 문제해결력 함양을 위한 방법 및 학습 전략에 대한 많은 연구가 필요하다. 또한 현재까지의 프로그래밍 학습은 대체로 개별 학습으로 이루어졌다. 프로그램의 문제 인식, 문제 분해에서부터 디버깅 과정까지 개인적으로 해결해가며 컴퓨팅 사고력을 함양하는 것이 주를 이루었다. 이러한 필요성 하에 2015 개정 교육과정과 OECD에서 공통적으로 중요시하는 미래인재의 역량인 협력적 문제해결력을 함양할 수 있는 실질적인 협업 도구와 교육 프로그램이 요구된다. 이러한 협업도구 및 프로그램은 학교 현장에서 개인의 문제해결력이 아닌 협업을 통한 협력적 문제해결력을 함양할 수 있는 실제적이고 체계적인 방법이어야 한다. 이를 위하여 구체적인 협업학습전략인 짝프로그래밍(Pair Programming)을 기반으로 SW교육 프로그램을 제안하고 학생들의 협력적 문제해결력을 효과적으로 증진시키기 위한 협업도구를 개발하여 수업에 적용하고자 한다.

2. 이론적 배경

2.1 SW교육과 협력적 문제해결력

SW교육은 궁극적으로 컴퓨팅 사고력을 갖춘 창의·융합적 인재 양성을 위한 교육으로서 SW교육과 가정 밀접한 2015 정보 교육과정에 따르면 컴퓨팅사고력과 더불어 정보문화소양과 협력적 문제해결력을 함께 함양해야 하는 핵심역량으로 제시하고 있다[12].

미래사회에서 타인과 더불어 살아가는 공동체 역량, 관계형성 능력, 의사소통역량 등의 핵심 역량들을 신장시키기 위하여 교육의 흐름은 학생중심 수업, 협력학습, 프로젝트 학습 등 학생간의 상호작용을 증시하는 방향으로 나아가고 있다. 이에 따라 ATC21S(Assessment & Teaching of 21st Century Skills)는 미래 사회에 필요한 핵심 소양으로서 협력적 문제 해결력을 강조하며 학생이 스스로 모둠을 조직, 분업하고 협력적으로 문제를 해결하는 과정을 평가하고 있다[4]. ATC21S는 협력적 문제 해결력을 개인의 사회적 기능과 인지적 기능이 주어진 상황에 따라 동시에 복합적으로 발휘되는 능력으로 간주하여 사회적 기능과 인지적 기능의 하위 기능들을 지표에 따라 학생들의 역량을 ‘낮음, 중간, 높음’의 3단 평가로 분석하는 평가틀과 평가요소를 설정하였다[15].

2.2 짝프로그래밍

짝프로그래밍은 하나의 컴퓨터에서 두 명의 프로그래머가 함께 작업하는 방식을 말한다[7][8][9]. 짝프로그래밍을 할 때, 두 명의 프로그래머는 나란히 앉아 코드를 설계, 알고리즘, 코딩, 디버깅을 협력하며 완성해나간다. Driver 역할은 컴퓨터의 키보드와 마우스로 작업하며 실제 코드를 설계하고 작성한다. Navigator의 역할은 Driver와 함께 작업을 보완하며 코드의 문법적인 오류, 오타 등을 리뷰해주며 코드의 전략이나 방향성에 대해 필요한 알고리즘을 설명한다[5][6].

짝프로그래밍의 교육적 효과에 대한 선행연구[1][2][3][10][14]들을 살펴보면 공동 개발의 결과물에 대한 질적 향상이 이루어짐과 동시에 부가적인 긍정적인 면도 있었다. 특히, 대부분의 논문에서 학생들의 만족감과 자신감을 높이는 결과를 보였다. 또한 짝프로그래밍을 통해 남학생보다는 여학생의 프로그래밍에 대한 태도에 많은 도움을 주는 것으로 검증되었다. Cockburn,

Williams는 작프로그래밍의 비용에 대해 연구한 결과, 프로그램 개발의 질뿐만 아니라 결점 감소, 개발자의 기술력 증가, 의사소통능력이 향상되었으며 작업 만족도도 증가한 것으로 나타났다[13][16].

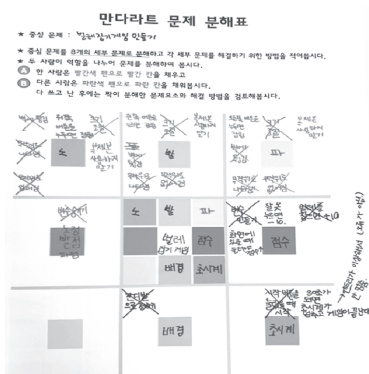
3. 작프로그래밍 협업도구를 활용한 SW교육프로그램의 개발

3.1 작프로그래밍을 위한 협업도구의 개발 및 활용 방안

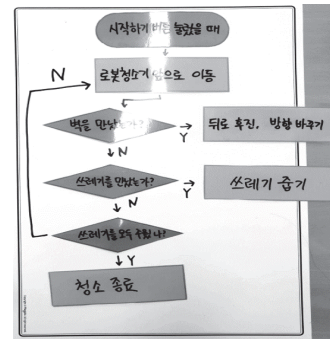
3.1.1 만다라트 문제 분해표

만다라트 문제 분해표는 프로그래밍 과정 중 가장 첫 번째 단계인 문제 인식 및 분해 단계에서 작과 함께 효율적인 의사소통을 위해 활용될 수 있는 협업 도구이다. 만다라트 분해표는 만다라트 발상법을 기반으로 문제 분해를 위해 제작된 협업도구이다. 만다라트 발상법이란 1개의 목표 달성을 위하여 8개의 주제를 설정하고 8개의 주제 실현을 위해 총 64개의 아이디어를 구체화하는 아이디어 발상법이다. 만다라트 발상법은 의사소통을 위한 도구로서 다양한 연구에서 활용되고 있다.

[Fig. 1]과 같이 만다라트 문제분해표는 총 81개의 칸으로 구성된 칸으로서 가운데에 핵심 과제 즉, 중심 문제를 적어놓고 각 색깔이 칠해져 있는 칸 8개에 세부 문제를 분해하고 마지막 노란색 칸에 각 세부문제를 해결하기 위한 실행 방법을 정리하는 브레인스토밍 틀이다.



[Fig. 1] Using Mandalart Problem Decomposition Table



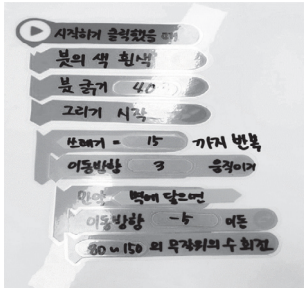
[Fig. 2] Flowchart magnetic board

3.1.2 순서도 자석 퍼즐

순서도란 다이어그램 중 하나로 처리 순서를 기호로 단계별로 나타낸 것을 말한다. 약속된 기호와 흐름선을 활용하여 명령 순서를 쉽게 알 수 있으며, 알고리즘을 설계하고 학습함에 있어 효율적이다[13]. 프로그램은 연속적인 단계로 이루어져 있으므로 전체적인 흐름을 파악하고 체계적인 처리절차를 계획하는 것이 중요하다. [Fig. 2]는 순서도 보드 블록의 활용 예시로서 순서도 보드 블록은 학생들에게 어려울 수 있는 순서도 작성을 쉽게 도와줄 수 있는 교구이다. 작과 함께 사용할 수 있는 화이트보드에 초등에서 자주 사용하는 기능과 관련된 순서도 기호를 블록으로 제시하고 이의 순서를 토의하며 흐름을 이해할 수 있도록 하였다. 하나하나의 지엽적인 코드 블록에 집중하는 것이 아니라 전체적인 흐름 파악을 통해 코드 작성에 있어 어려움을 줄여줄 것이다.

3.1.3 코드 블록 자석

어린 학습자들이 프로그래밍의 개념이나 언어를 학습하는 것은 어려움이 따를 수밖에 없다. 그러므로 초등학교 급에서 이루어지는 SW교육에서 학습자들의 발달 특성에 따라 적합한 교구나 구체적 조작물을 사용하는 것이 중요하다. 이러한 SW교육에서의 구체적 조작물의 필요성에 따라 코드 블록 자석은 실제 프로그래밍을 시작하기 전 작과 의견 교환의 도구로서 사용하거나 프로그래밍 도중 수정사항을 이야기할 때 구체적인 의견 표현 수단으로서 학습자들에게 활용될 수 있다. <Fig. 3>은 코드 블록 자석의 활용 예시로서 디버깅 과정에서 오류를 찾아내려고 할 때 각 코드 블록들을 작은 단위로 분해하고 순서를 재조정하는 과정에서 효율적으로 활용될 수 있다.



[Fig. 3] Code block magnet

3.1.4 디버깅 체크리스트

디버깅이란 프로그래밍 과정 중에서 생기는 오류를 해결하는 것을 말한다. 프로그래밍 수업 도중 오류가 발생하게 되면 원인은 무엇이고 해결 방법은 무엇인지 파악하기 어려운 난관에 봉착하게 되는 경우가 굉장히 많다. 이를 위하여 학생들의 협력적인 디버깅을 위한 스캐폴딩 장치로서 디버깅 체크리스트를 개발하였다. 체크리스트는 짝프로그래밍에서 드라이버와 내비게이터가 공통적으로 활용할 수 있게끔 구성되었다. 디버깅 전략의 순서와 프로그램에서 오류를 찾아내는 구체적인 8가지 방법이 제시되어있으며 블록 코드에서 자주 일어나는 프로그램 오류를 문법 오류, 논리오류로 나누어 검증할 수 있도록 하였다. [Fig. 4]의 체크리스트와 같이 학생들은 프로그래밍의 논리오류에 봉착하였을 때 오류의 원인을 탐색하는 방법에 따라 순서대로 전략을 수행하며 이를 원하는 방향으로 수정하는 방법에 대한 스캐폴딩을 지원받고 짝과의 상호작용을 통해 손쉽게 디버깅을 할 수 있다.

3.2 교육 프로그램의 개발 방향 및 내용

3.2.1 교육 프로그램 개발 방향

본 교육 프로그램은 프로그래밍 언어와 문법에 대한 지식, 프로그래밍 기법에 대해 배우는 것에 초점이 맞춰져 있기 보다는 짝프로그래밍을 통해 협력적 문제해결력을 함양하는데 중점을 두고 있다. 그러므로 SW교육을 처음 접한 학생들을 대상으로 하는 것이 아닌 일반적인 블록코드를 경험해보고 기능들을 기본적으로 이해하고 실행할 수 있는 학생들을 대상으로 한다. 다음의 교육 프로그램에서는 교사 설명식의 지식 전달 수업은 지양하며 교사는 학생들의 협력 과정 속에서 생기는 마찰을 조정해주는 역

Pair Programming을 위한 디버깅 체크리스트

1. 디버깅이란?
프로그래밍에서 잘못된 부분(오류)을 찾아 고치는 것
고치는 것에서 더 나아가 더 나은 코딩, 더 효율적인 코딩을 하는 것
근거적인 디버깅 순서와 전략 (모든 전략을 꼭과 함께 수행하여야 하지만, 각 전략을 주도적으로 수행해야 할 역할에 생일이 일해서 있습니다. 디버깅 전략을 꼭과 순서대로 실천하며 체크해봅시다.)

디버깅 전략	드라이버	내비게이터
1. 파일 남김을 방지하기 위해 오류가 발생하는 원본 프로젝트에 대한 복사본을 미리 만들어 놓고 복사본으로 오류 수정 준비하기		
2. 처음부터 사물레어를 사용하여 오류가 발생한 즉시 프로그램 중단하기		
3. 프로그램에서 오류 찾아내는 방법		
①오류가 의심 가는 지점을 직접 수정하기		
②오류가 발생한 부분을 모두 제거한 뒤 오류가 있는 코드 블록을 하나씩 추가하며 찾아내기		
③오류가 발생한 부분에서 코드블록을 하나씩 제거하며 찾아내기		
④오류가 의심되는 코드를 하나씩 매서시 순서대로 실행시켜보기		
⑤발생하는 오류를 눈에 띄게 만들기 위해 오류의 실행 오류를 크게 하기(예: 반복 블록을 이용하여 특정 오류 반복시키기, 특정 코드의 수직(수차 값)을 증가시키기)		
⑥발생하는 오류의 비슷한 코드를 똑같이 만들어보기		
⑦무엇이 오류를 발생시키는지 알려주도록 주석해고고 가정을 세워보기		
⑧불필요한 코드 블록은 없는 지 확인하기		
최대한 코드 블록은 단순하게 만들기		
4. 오류 해결의 우선 순위 정하기 (해결해야 할 문제를 실수로 파악하지 않을 수 있으며 자주 발생하는 오류를 정리할 수 있다.)		
순위	오류	원하는 방향
1		
2		
3		
5. 오류를 찾아낸 후 내가 의도한 바가 맞는 지 다시 검증하기		
블록코딩에서 자주 발생하는 오류(체크리스트 점검을 통해 오류를 최소화해봅시다.)		
점검 문항	체크	
1.오류가 있지는 않는가? 문법 오류		
2.잘못된 형식의 블록을 끼워 넣지는 않는가? (변수, 리스트, 숫자값, 문자열 등) 논리 오류		
3.조건이 여러 개라면 앞의 조건과 뒤의 조건의 순서가 바뀌어 하지는 않는가? (조건 순서의 문제) 조건 설정의 오류		
4.'말일-라벤'의 조건을 설정하였을 때 조건이 참 또는 거짓의 조건 설정이 적절인가? (알려져야 할 조건이 숫자 값일 경우 그 숫자는 정확한가? 하나 더 크거나 작아야 하는 것은 아닌가?) 변수 설정의 오류		
5.변수가 있다면 변수의 범위가 적절인가?		
6.조건이나 반복이 실행되는 중에도 변수는 특정한 값을 이미 가지고 있다. 어느 흐름에서 변수가 어떤 값을 가지고 있는 지 확인해보자.		
7.무작위의 값을 추출한다면 무작위 값의 범위는 적절인가? 반복하기 주위의 오류		
8.반복하기 블록을 사용해야 하는가, 무반복하기 블록을 사용해야 하는가?		
9.반복하기 블록을 사용할 때 처리 속도를 줄이기 위한 '기다리기' 블록을 활용할 필요가 있는가? 반복하기 블록의 범위는 어디까지 지정해야 하는가?		
10.반복하기 블록을 사용하였을 때 실행 흐름을 잘못 설정하여 실행이 실행할 필요가 있는가? 실행 흐름의 오류		
11.'실행 흐름' 블록을 사용하였을 때 실행 흐름을 잘못 설정하여 실행이 어긋나지는 않았는가? 문법 설정의 오류		
12.유지되는 회전 블록을 사용할 때 오브젝트 모양의 이동방향을 확인하였는가? 다른 프로그램 참고하기		
13. 다른 프로젝트의 정상적인 코드와 내 프로그램에서 문제가 발생한 코드를 비교하여 차이점을 들여보자.		
발생한 오류	해결 방법	
1		
2		
3		

[Fig. 4] Debugging checklists

할을 맡아 최대한 학생의 상호작용이 원활하게 일어나게끔 하여 협력적 문제해결력을 함양하는데 중점을 두었다.

3.2.2 교육 프로그램 개발 내용

짝프로그래밍을 진행하기 위한 교육 활동은 3단계로 이루어진다. '준비하기'단계에서는 짝프로그래밍 수행을 위한 안내와 역할 소개, 역할 분담 그리고 짝을 배정하는 단계이다. 이 때 가장 중요한 것은 짝을 배정하는 것으로 짝 사이에 수준 차이가 많이 나게 되면 수준 높은 학생이 낮은 학생을 가르치는 식의 과외처럼 진행되게 되므로 최대한 수준이 비슷한 학생들로 선정해야 하는 것이 중요하다.

‘협력 실습하기’단계에서는 드라이버와 내비게이터로 나누어 맡은 역할에 따라 협력적으로 프로그래밍을 한다. 이 때 한 명의 드라이버 혹은 내비게이터의 독단이나 주도하에 프로그래밍이 진행되지 않도록 주의하는 것이 중요하다. 이를 위해 10분에 한 번씩 두 역할을 교환하는 것으로 한 역할에게 많은 주도권이 가지 않도록 적절히 분배되어야 한다. 또한 이러한 협력 과정에서 토의한 내용이 간단하면서도 체계적으로 구조화되어야 실제 프로그래밍 과정이 효율적으로 진행될 수 있다. 이를 위해 체크리스트 등의 협업 도구가 활용될 수 있다.

마지막으로 ‘되돌아보기’ 단계에서는 작성한 코드에 대해 짝과 함께 효율성에 대해 평가해보고 개선할 수 있는 방안에 대해 토의한다. 이러한 과정에서 교사는 학생들의 협업과정을 관찰평가하며 다음 작프로그래밍을 위한 짝 선정의 근거자료를 준비할 수 있다.

다음으로 작프로그래밍 교육 프로그램의 차시별 활동 및 각 차시에서 활용 가능한 협업 도구에 대해 살펴보면 다음과 같다. 1차시에서는 작프로그래밍에 대한 안내와 관련 역할에 대해 알아보고 수준 진단을 통해 짝을 선정한다. 2~5차시는 작프로그래밍에서 프로그래밍의 단계 수행을 각 협업 도구를 활용하여 효율적으로 진행한다. 6~15차시에서는 4가지 협업도구를 모두 활용하여 하나의 프로젝트를 완성할 수 있는 프로그래밍 미션을 해결하게 된다. 본 교육 프로그램에서는 8차시에 걸친 실습만 제시하였지만 상황과 예제 프로그램의 수에 따라 차시를 유동적으로 증감시킬 수 있다. 마지막 16차시에서는 작프로그래밍 활동을 평가해보며 바람직한 드라이버와 내비게이터의 역할과 상호작용을 통한 문제해결을 위한 노력에 대해 토의하며 협력적 문제해결능력이 함양될 수 있도록 한다.

4. 연구 실험 설계 및 결과

4.1 연구 실험 설계

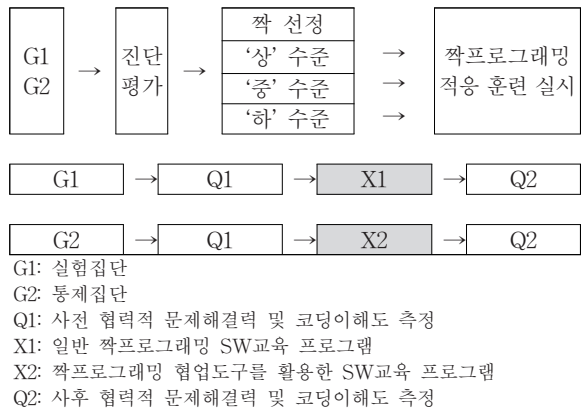
본 연구는 초등학교 5~6학년으로 구성된 소프트웨어 동아리 학생들을 대상으로 방과 후 동아리 시간을 활용하여 작프로그래밍 소프트웨어 교육을 진행하였다. 학생들의 협력적 문제해결력 및 코딩 이해도에 대한 효과성을 검증하기 위하여 협업 도구를 활용한 교육 프로그램

를 개발하고, 개발된 프로그램을 단일 집단에 적용한다. 작프로그래밍 SW교육을 실시하였을 경우와 협업 도구를 활용하여 작프로그래밍 SW교육을 실시하였을 경우의 결과 차이를 비교·분석한다.

실험 전 실험 집단은 진단평가를 실시한 뒤 진단평가의 결과에 따라 학생들의 코딩 수준을 상, 중, 하로 나눈다. 같은 코딩 능력 수준을 가진 학생끼리 동질 집단의 짝을 구성한다. 본 연구는 작프로그래밍의 효과성에 관한 연구가 아닌 작프로그래밍 학습 시 협업도구 활용의 효과성에 주안점을 둔 연구이므로 각 짝 그룹들을 작프로그래밍을 능숙하게 다룰 수 있도록 적응훈련 기간을 둔다. 그 후 통제집단의 학생에게는 협업 도구를 활용하지 않는 작프로그래밍 SW교육을 실시한다. 작프로그래밍을 통해 5~6학년 중급 수준의 미션 작품 2가지를 해결하도록 제시한 뒤 해결 과정에서의 협력적 문제해결력 및 코딩 이해도를 측정한다.

실험 집단의 학생은 작프로그래밍 협업도구를 활용한 SW교육(표 5에서의 X2)을 실시한다. 협업 도구를 활용하여 5~6학년 중급 수준의 미션 작품 3가지를 해결하도록 제시한 뒤 해결 과정에서의 협력적 문제해결력 및 코딩 이해도를 측정한다.

수업 처치 후에는 협업 도구를 사용하지 않은 교육 프로그램 적용에 대한 사후검사와 협업 도구를 사용한 교육 프로그램 적용에 대한 사후 검사의 차이를 분석 및 해석한다. 본 연구에서 실시하는 실험 설계는 <Fig. 5>와 같다.



<Fig. 5> Experiment design

4.2 검사 도구 개발

4.2.1 짝프로그래밍 협업도구를 활용한 SW교육에서의 협력적 문제해결력 검사 도구 개발

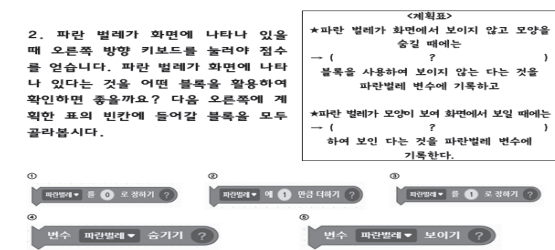
선행연구인 OECD, 2013; ACT2S 2012;에서 다루었던 협력적 문제해결력 평가 도구들의 기본 평가틀과 평가문항들을 초등학생 대상 짝프로그래밍 SW교육 프로그램의 개발 방향에 맞게 일부 수정 및 조합하여 활용하였다.

초등 SW교육에서의 협력적 문제해결력 검사 도구의 기본적인 평가틀은 ATC21S의 협력적 문제해결력 평가틀을 바탕으로 제작하였다. 짝프로그래밍 과정 중 일어날 수 있는 협력적 문제 해결의 인성적인 측면과 사회적 기술에 관한 문항으로 구성하되 선행연구의 협력적 문제해결력 평가 문항을 짝 프로그래밍 교육 프로그램의 흐름에 맞게 용어와 내용을 일부 수정하였다.

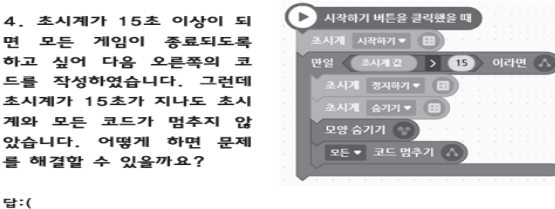
4.2.2 코딩 이해도 측정 도구 개발

짝프로그래밍 협업도구를 활용한 SW교육 프로그램을 통해 학생들이 코딩한 프로그램에 대한 이해도를 측정하기 위한 간단한 평가 문항을 개발하였다. 평가 문항의 유형은 크게 3가지로 다음의 그림과 같다.

‘코드 블록 기능 이해하기’의 문항은 특정한 상황에서의 기능을 구현할 때의 블록을 선택하고 특정 개념을 이해하고 있는가를 평가하며 문항 예시는 [Fig. 6]과 같다.



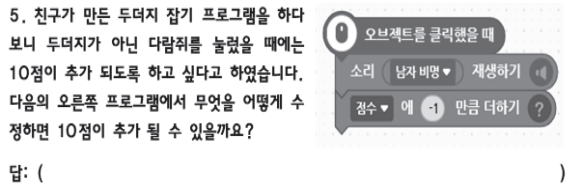
[Fig. 6] Code block function understanding



답: ()

[Fig. 7] Code error debugging quiz

‘코드 오류 디버깅하기’ 문항은 코드를 작성하는 중 범하기 쉬운 오류를 제시하고 이를 디버깅하는 유형으로서 문항 예시는 [Fig. 7]과 같다.



[Fig. 8] Project remaker quiz

‘프로젝트 리메이크’ 문항은 기존의 과제로 완성한 프로젝트를 다른 방향으로 리메이크할 시에 어떠한 방법으로 코드를 작성할 수 있는 지에 대해 묻는 유형으로 문항 예시는 [Fig. 8]과 같다.

4.3 연구 결과 분석

4.3.1 협력적 문제해결력 분석

짝프로그래밍 협업도구를 활용한 SW교육 프로그램이 학생들의 협력적 문제해결력에 미치는 영향을 분석하기 위하여 협력적 문제해결력 측정을 실험집단과 통제집단을 대상으로 프로그램 적용 전과 후에 실시하였다. 실시한 후의 모든 자료는 R 통계 프로그램(ver. 3.4.1)을 활용하여 T검정을 실시하였으며 결과 값은 소수 둘째자리까지 반올림하여 나타내었다.

4.3.1.1 사전 실험집단과 통제집단의 협력적 문제해결력 비교

사전 두 집단의 협력적 문제해결력 차이가 있는 지 확인하기 위해 독립표본 T검정을 한 결과 <표 1>과 같

<Table 1> Independent sample T-test (Pre-Test) of two groups

Result	Area	Cognitive function	Social function	Total
Control Group	M	67.80	80.40	148.20
	SD	7.67	6.74	13.77
Experiment Group	M	71.00	76.60	147.60
	SD	4.42	3.78	7.88
t		-1.14	1.56	0.12
p		0.27	0.14	0.91

이 유의수준 0.05에서 통계 값은 0.12, 유의확률은 0.91로 실험집단과 통제집단 간의 사전 협력적 문제해결력은 유의한 차이가 없었다. 그러므로 프로그램을 적용하기 전의 두 집단 간의 협력적 문제해결력은 동질하다고 판단할 수 있다.

4.3.1.2 사후 실험집단과 통제집단의 협력적 문제해결력 비교

작프로그램밍 SW교육 프로그램을 적용한 후 집단 간의 협력적 문제해결력 차이를 T검정을 통하여 알아본 결과 <표 2>와 같이 실험 집단의 평균은 170.10, 표준편차는 8.79이고, 통제집단의 평균은 190.40, 표준편차는 4.84로 나왔다. 통계값은 -6.40, 유의확률은 0.00로서 유의수준 0.05에서 사후 실험집단과 통제집단 사이에는 협력적 문제해결력이 유의한 차이가 있는 것으로 나타났다. 그러므로 작프로그램밍 협업도구를 활용한 SW교육프로그램이 협업도구를 활용하지 않았을 때보다 학생의 협력적 문제해결력을 신장시키는데 유의한 효과가 있었음을 알 수 있다.

<Table 2> Independent sample T-test (Post-Test) of two groups

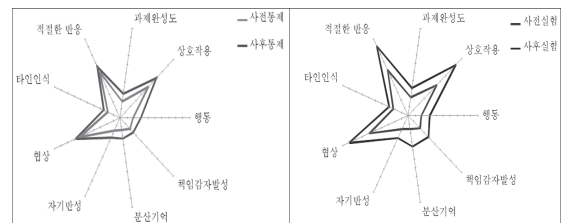
Result	Area	Cognitive function	Social function	Total
Control Group	M	77.60	92.50	170.10
	SD	6.02	4.03	8.79
Experiment Group	M	89.40	101.00	190.40
	SD	2.12	3.06	4.84
t		-5.85	-5.31	-6.40
p		0.00	0.00	0.00

<표 3>은 사후 실험집단과 통제집단의 영역별 독립 표본 T검정 결과를 표로 나타낸 것이다. 표와 같이 사회적 기능 영역에서는 상호작용, 적절한 반응, 협상, 분산기억, 책임 진취성 영역에서 실험집단과 통제집단 간의 유의한 차이가 있는 것으로 나타났다. 인지적 기능 영역에서는 전 영역에서 실험집단과 통제집단 간의 유의한 차이가 있는 것으로 나타났다.

[Fig. 9]는 실험집단과 통제집단의 협력적 문제해결력 중 사회적 기능 영역 평균 점수의 사전, 사후 변화를 그래프로 나타낸 것이다. [Fig. 9]의 왼쪽은 통제집단, 오른쪽은 실험집단의 그래프이다. 그래프를 통해 학생들의 코딩 이해도의 평균 증가폭은 작프로그램밍 협업도

<Table 3> Independent sample T-test (Post-Test) of two groups per area solving ability in social function area

Area	Result	Control		Experiment		t	p
		M	SD	M	SD		
Social Function	behavior	8.60	0.70	8.30	0.95	0.80	0.43
	interaction	14.70	1.16	16.50	1.35	-3.19	0.01
	task completion	8.30	0.67	8.40	0.70	-0.33	0.75
	proper response	14.40	2.72	16.70	1.06	-2.49	0.03
	awareness	8.00	0.47	8.10	0.88	-0.32	0.76
	negotiations	14.80	1.32	16.60	0.84	-3.64	0.00
	self-assessment	8.00	0.67	8.10	0.32	-0.43	0.68
	distributed memory	7.70	0.95	9.00	0.82	-3.28	0.00
	initiative of responsibility	8.00	0.67	9.30	0.67	-4.33	0.00
	total	92.50	4.03	101.00	3.06	-5.31	0.00
Cognitive Function	task analysis	8.00	0.82	9.60	0.52	-5.24	0.00
	target Settings	8.00	0.82	9.10	0.74	-3.16	0.01
	resource utilization	7.70	0.95	8.90	0.57	-3.43	0.00
	flexibility	7.70	0.82	8.50	0.71	-2.33	0.03
	gathering information	7.60	0.97	8.70	0.48	-3.22	0.01
	systematization	7.50	0.53	8.70	0.67	-4.43	0.00
	identification of relevance	7.90	0.74	9.10	0.74	-3.64	0.00
	if rule	7.70	0.67	8.70	0.67	-3.31	0.00
	Assume	15.50	1.43	18.10	0.88	-4.89	0.00
	Total	77.60	6.02	89.40	2.12	-5.85	0.00
overall cooperative problem-solving ability	170.10	8.79	190.40	4.84	-6.40	0.00	

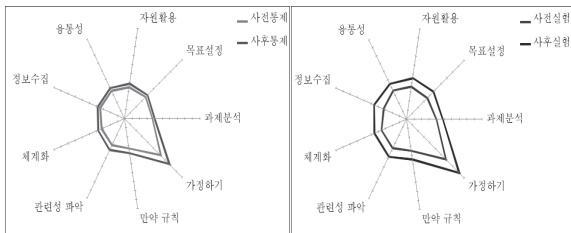


[Fig. 9] Comparison of collaborative problem

구를 활용한 SW교육프로그램을 실시한 실험집단의 평균 점수가 더 높음을 확인할 수 있다.

[Fig. 10]은 실험집단과 통제집단의 협력적 문제해결력 중 인지적 기능 영역 평균 점수의 사전, 사후 변화를 그래프로 나타낸 것이다. [Fig. 10]의 왼쪽은 통제집단, 오른쪽은 실험집단의 그래프이다. 그래프를 통해 학생들의 코딩 이해도의 평균 증가폭은 작프로그램밍 협업도구를 활용한 SW교육 프로그램을 실시한 실험집단의 평균

점수가 더 높음을 확인할 수 있다. 그러므로 짝프로그래밍 협업도구를 활용한 SW교육프로그램은 학생의 협력적 문제해결력 중 인지적 기능 영역과 사회적 기능 영역을 신장시키는 데 효과적인 것으로 판단할 수 있다.



[Fig. 10] Comparison of collaborative problem solving ability in cognitive function area

4.3.2 코딩 이해도 분석

4.3.1.1 실험집단과 통제집단의 사전 코딩 이해도 비교

프로그램 적용 전에 실험집단과 통제집단 간의 코딩 이해도에 대한 차이를 알아보기 위해 R 통계 프로그램(ver. 3.4.1)을 활용하여 독립표본 T검정을 실시하였다. 결과는 <표 4>과 같으며 모든 결과 값은 소수 둘째자리까지 반올림하여 나타내었다.

<Table 4> Independent sample T-test (Pre-Test) of two groups

Result \ Group	Experiment	Control
M	68.00	70.00
N	10	10
t		-0.25
p		0.81

사전 두 집단의 코딩 이해도 차이가 있는 지 확인하기 위해 독립표본 T검정을 한 결과 유의수준 0.05에서 통계 값은 -0.25로 실험집단과 통제집단 간의 사전 코딩 이해도에 유의한 차이가 없는 것으로 나타났다. 그러므로 프로그램을 적용하기 전의 두 집단 간의 협력적 문제해결력은 동질하다고 판단할 수 있다.

4.3.1.2 실험집단과 통제집단의 사후 코딩 이해도 비교

짝프로그래밍 SW교육 프로그램을 적용한 후 실험집단과 통제집단 간의 코딩 이해도 차이를 알아보기 위해 R 통계 프로그램(ver. 3.4.1)을 활용하여 독립표본 T검정을 실시하였다. 결과는 <표 5>과 같으며 모든 결과 값은 소수 둘째자리까지 반올림하여 나타내었다.

단과 통제집단 간의 코딩 이해도 차이를 알아보기 위해 R 통계 프로그램(ver. 3.4.1)을 활용하여 독립표본 T검정을 실시하였다. 결과는 <표 5>과 같으며 모든 결과 값은 소수 둘째자리까지 반올림하여 나타내었다.

<Table 5> Independent sample T-test (Post-Test) of two groups

Result \ Group	Experiment	Control
M	89.00	76.00
N	10.00	10.00
t		-2.70
p		0.02

짝프로그래밍 SW교육 프로그램을 적용한 후 집단 간의 코딩 이해도 차이를 T검정을 통하여 알아본 결과 실험 집단의 평균은 89.00이고, 통제집단의 평균은 76.00으로 나왔다. t통계값은 -2.70, 유의확률은 0.02로서 유의수준 0.05에서 사후 실험집단과 통제집단 사이에는 코딩 이해도가 유의한 차이가 있는 것으로 나타났다. 그러므로 짝프로그래밍 협업도구를 활용한 SW교육 프로그램이 협업도구를 활용하지 않았을 때보다 학생의 코딩 이해도를 신장시키는데 유의한 효과가 있었음을 알 수 있다.

4.3.1.2 실험집단과 통제집단의 코딩 이해도 향상도 비교

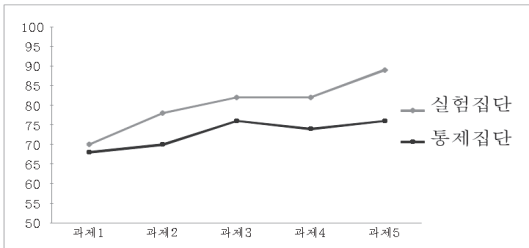
통제집단과 실험집단이 5차례의 짝프로그래밍 과제를 실시하는 과정에서 코딩 이해도의 평균 점수 변화는 <표 6>과 같다. 협업도구를 활용하지 않은 일반 짝프로그래밍 SW교육프로그램을 실시한 통제 집단은 과제1에서 과제5에 이르기까지 최고 8점의 점수가 향상되었지만, 협업도구를 활용하여 짝프로그래밍 SW교육프로그램을 실시한 실험집단은 최고 70점에 89점까지 19점의 점수가 향상된 것을 알 수가 있다.

<Table 6> Average coding scores per report

Group \ Report	No.1	No.2	No.3	No.4	No.5
Control	68	70	76	74	76
Experiment	70	78	82	82	89

[Fig. 11]은 실험집단과 통제집단의 과제별 코딩 이해도의 평균 점수를 그래프로 나타낸 것이다. 그래프를 보았을 때 학생들의 코딩 이해도의 평균 증가폭은 짝프로

그래밍 협업도구를 활용한 SW교육 프로그램이 더 높았음을 알 수 있다. 그러므로 썩프로그래밍 협업도구를 활용한 SW교육 프로그램은 학생의 코딩 이해도를 신장시키는 데 효과적인 것으로 판단할 수 있다.



[Fig. 11] Coding scores comparison per report

5. 결론 및 향후 연구 과제

본 연구는 썩프로그래밍 협업도구를 활용한 SW교육 프로그램을 적용 및 개발하는데 목적이 있다. 이러한 목적을 달성하기 위하여 협력적 문제해결력을 함양할 수 있도록 4가지의 협업도구를 개발하였다. 즉, 프로그래밍의 문제 인식 및 분해 단계에서 활용하는 만다라트 문제 분해표, 알고리즘 이해 단계에서 활용하는 순서도 자석 퍼즐, 알고리즘 표현 단계에서 활용하는 코드 블록 자석, 알고리즘 분석 및 검증 단계에서 활용하는 디버깅 체크리스트를 개발하였으며 이를 활용한 SW교육 프로그램을 초등학교 수업에 적용하였다.

프로그램 적용 후 다음과 같은 결과를 얻었다.

첫째, 실험 집단 간의 협력적 문제해결력 차이를 T검정을 통하여 알아본 결과, 사후 실험집단과 통제집단 사이에는 협력적 문제해결력이 유의한 차이가 있었다. 그러므로 썩프로그래밍 SW교육 프로그램에서 협업도구의 유무가 협력적 문제해결력 신장 정도에 유의한 효과를 보였다.

둘째, 사후 집단 간의 코딩 이해도 차이를 T검정을 통하여 알아본 결과, 실험집단과 통제집단 사이에는 코딩 이해도가 유의한 차이가 있었다. 그러므로 썩프로그래밍 SW교육 프로그램에서 협업도구의 유무가 코딩 이해도에 유의한 효과를 보였다.

본 연구는 썩프로그래밍 협업도구를 활용한 SW교육 프로그램을 개발 및 적용하여 효과성을 밝힌 데 의의가 있다. 초등학생들의 협력적 문제해결력 함양을 위한 협업 도구 혹은 협력적 SW교육프로그램에 대한 더욱 다양하고 체계적인 후속 연구가 이루어지길 기대한다.

참고문헌

- [1] Brian Hanks (2006), Student attitudes toward pair programming, *ITiCSE 2006*, pp. 113-117.
- [2] Charlie McDowell et al. (2006), Pair programming improves student retention, confidence, and program quality, *CACM 49*(8), pp. 90-95.
- [3] Charlie McDowell, Linda L. Werner, Heather E. Bullock, Julian Fernald (2002), The effects of pair-programming on performance in an introductory programming course, *SIGCSE 2002*, pp. 38-42.
- [4] Griffin, P. E., Care, E. (2015), *Assessment and teaching of 21st century skills*, New York : Springer.
- [5] Hyo Won Noh, Jin Ho Park, Hoon Sung Gwak. (2013), The Design, Implementation and Verification of Distributed Pair Programming System for Supporting Collaboration, *The journal of Korea Navigation Institute*, 17(3), 346-353.
- [6] Keun Woo Han, Eun Kyoung Lee, Young Jun Lee. (2006). The Effects of Pair Programming on Achievement and Motivated Strategies in Programming Course, *The Journal of Korean association of computer education*, 9(6), 19-28.
- [7] Laurie A. Williams et al. (2000), Strengthening the Case for Pair Programming, *IEEE Software* 17(4), pp. 19-25.
- [8] Laurie A. Williams, Richard L. Upchurch (2001), In support of student pair-programming, *SIGCSE 2001*, pp. 327-331.
- [9] Laurie Williams, Eric N. Wiebe, Kai Yang, Miriam Ferzli, Carol Miller (2002), In Support of Pair Programming in the Introductory Computer Science Course, *Computer Science Education* 12(3), pp. 197-212.

- [10] Linda L. Werner, Brian Hanks, Charlie McDowell, Pair-programming helps female computer science students, *ACM Journal of Educational Resources in Computing* 4(1), Article No. 4.
- [11] Min Huh, Tae-Wuk Lee (2014). Exploration of Information Subject-centered Curriculum Integration Strategies for 21 Century Key Competencies Extension, *Journal of the Korea society of computer and information*, 19(2), 253-261.
- [12] Ministry of Education (2015). *Practical course/Informatics Curriculum, Sejong: Ministry of Education.*
- [13] Myoung-sub Song, Youngsik Kim. (2018). Comparison of Flow Chart Before and After Programming, *The Korean Association of Computer Education*, 23(1), 195-198.
- [14] Nachiappan Nagappan et al. (2003), Improving the CS1 experience with pair programming, *SIGCSE 2003*, pp. 359-362.
- [15] Park, Hye Young, Rim, Haemee. (2014). Analyzing features of collaborative problem solving competencies in PISA and ATC21S : Implications for instruction and assessment in Korea, *Korean Association for Learner-centered Curriculum and Instruction*, 14(9), 439-462.
- [16] So Eun Jeon, So Jung Choi, Woojin Paik. (2008). Evaluating the Effectiveness of Pair Programming, *Korean Society for Information Management*, 201-206.

저자소개

김 용 옥



2017~ 서울교육대학교 대학원 컴퓨터교육전공 석사과정
 2016~ 현재 서울 서래초등학교 교사
 관심분야 : 컴퓨터교육, SW교육, 3D 프린팅 교육
 e-mail : cindy6242@sen.go.kr

전 석 주



2002 한국과학기술원 컴퓨터 공학 박사
 2004 ~ 현재 서울교육대학교 컴퓨터교육과 교수
 관심분야 : 컴퓨터교육, 프로그래밍방법, 데이터마이닝, 멀티미디어DB
 email : chunsj@snue.ac.kr