

초등학생의 프로그래밍 학습에서 활동지를 사용한 성찰에 대한 사례 연구

김용천* · 김자미** · 이원규***

고려대학교 컴퓨터교육학과* · 고려대학교** · 고려대학교 컴퓨터교육과***

요약

최근 프로그래밍 학습 과정에서 학습자가 느끼는 인지적인 부담을 감소시키기 위한 방법으로 성찰에 대한 연구가 이루어지고 있다. 이에 본 연구에서는 프로그래밍 학습에서 활동지를 사용한 성찰 활동을 통해 학습자의 프로젝트 수행이 어떻게 달라지는지를 알아보기 위하여 초등학생을 대상으로 수업을 진행하였다. 연구 결과, 프로그래밍 학습에서 성찰의 시간이 많은 학습자는 다양하고 창의적으로 프로젝트를 수행하는 모습을 확인할 수 있었다. 본 연구는 초등학생의 프로그래밍 학습에서 시행착오를 통해 구현하기보다 스스로 생각하는 성찰을 통해 프로그래밍 학습을 할 수 있는 기회를 제공해주었다 데에 의의가 있다.

키워드 : 프로그래밍 학습, 활동지, 성찰

A Case Study on Reflection Using Worksheets for Elementary School Students in Programming Learning

Yong-Cheon Kim* · Ja-Mee Kim** · Won-Gyu Lee***

Dept. of Computer Science Education, Korea University*
Korea University*

Dept. of Computer Education, Korea University***

ABSTRACT

Recently, reflection as a way of reducing the learners' cognitive burden in the programming learning process has been studied. In the present study, we examined the effects of reflection using worksheets to measure elementary students' project performance. The results of the study are that learners who have more reflective time over their learning process are performed creatively and diversely their project work. This study is significant in that it provides more learning opportunities for elementary students by reflection rather than by trial-and-error in programming learning.

Keywords : Programming Learning, Worksheet, Reflection

이 연구는 2009학년도 고려대학교 사범대학 특별연구비 지원을 받아 수행되었음
논문투고: 2011-11-01
논문심사: 2011-11-04
심사완료: 2011-12-28

1. 서론

2007년 2월에 개정 고시된 중등 정보교과 교육과정은 문제 해결력 및 논리적 사고력을 향상시킬 수 있는 내용으로 개정되었다[1][4][7]. 개정교육과정에서는 ‘문제 해결 방법과 절차’ 영역에서 응용소프트웨어 활용 중심의 교육 내용을 축소하고 문제 해결력 향상을 위해 알고리즘적 사고 학습에 중점을 두어 지도할 것을 강조하고 있다[1]. 알고리즘적 사고 학습은 초등학교 단계에서부터 체계적인 프로그래밍 교육을 통해 가능할 것으로 보인다[9]. 알고리즘적 사고를 확장시킬 수 있는 프로그래밍 학습은 언어사용에서 제기되는 문법의 복잡성 등으로 인해 초등학생에게는 너무 어렵다는 문제가 제기되어 왔다[19][23][37]. 따라서 많은 연구들은 알고리즘을 컴퓨터로 구현할 때, 상대적으로 학습이 쉽고, 학습자의 흥미를 유발하는 데 기여하는 교육용 프로그래밍 언어의 사용을 권장하였다[3][9][15].

교육용 프로그래밍 언어는 기존의 프로그래밍 언어와 달리 비주얼 프로그래밍 방식의 인터페이스를 제공한다[26]. 교육용 프로그래밍 언어는 사용하기 쉽고 간단하기 때문에 자신이 작성한 알고리즘을 컴퓨터로 쉽게 구현해볼 수 있다[30]. 학습자는 교육용 프로그래밍 언어의 쉬운 사용으로 인해 문법 오류에 대한 부담에서는 벗어나게 되었다. 이상의 장점으로 인해 교육용 프로그래밍 언어는 고등교육의 프로그래밍 교육까지 사용 범위가 확대되었다[12][30].

교육용 프로그래밍 언어의 쉬운 사용에도 불구하고, 프로그래밍 학습은 논리적 사고 및 추상적 사고를 요구하기 때문에 여전히 학습자에게 과도한 인지적 부담을 유발할 수 있다[14][17][20]. 프로그래밍 학습에서 학습자가 느끼는 인지적인 부담을 감소시키기 위한 방법으로 성찰에 대한 연구가 이루어지고 있다. 초보 학습자는 시행착오를 통해 프로그래밍에 익숙해지지만, 디버깅이나 에러를 수정하기 위해서는 반성적 성찰이 요구된다[5]. 장윤재(2011)도 반성적 성찰을 통해 생각을 정리해 나가는 학습자가 그렇지 않은 학습자에 비해 프로젝트 완성 능력이 높은 것으로 보고하였다[12]. 프로그래밍 학습에서 성찰의 중요성에 대해 논의한 연구들도 성찰의 과정에서 학습자가 어떤 변화를 통해 프로그래밍을 쉽게 완성해 나가는지에 대해서는 고려하지 않았다. 즉, 성찰이 프로그래밍 학습에 필요한 과정임에도 불구하고, 반성적 사고와 성찰이 프로그래밍의 어떤 과정에 기여하는지에 대한 연구는 미흡하였다. 따라서 본 연구는 프로그래밍 학습에서 활동지를 활용한 성찰이 학습자의 프로그래밍 완성도를 높이는 데 어떻게 기여하고 있는지를 알아보기 위한 목적이 있다. 목적 달성을 위해 본 연구는 프로그래밍 과정에서 학습자의 성찰과 변화 과정에 대해 논의하였다.

2. 이론적 배경

2.1 프로그래밍 학습의 중요성

2007년 개정 정보교과 교육과정의 ‘문제 해결 방법과 절차’ 영역은 실생활의 문제를 해결하는데 도움을 줄 수 있는 내용으로 구성되었다. 문제 해결 영역은 논리적 절차인 알고리즘의 설계, 분석과 함께 다양한 방법으로 알고리즘을 구현해 봄으로써 학습 효과를 높일 수 있다는 것으로 정의하고 있다. 알고리즘을 컴퓨터로 구현하는 프로그래밍은 전문 프로그래머 양성을 위한 목적 뿐 아니라 다음과 같은 장점으로 인해 기초적인 프로그래밍 학습이 필요하다라는 주장이 제기되었다[10]. 첫째, 프로그래밍 학습은 학습자의 창의력, 문제 해결력, 논리적 사고력을 향상시켜준다[8][11]. 둘째, 프로그래밍을 통한 문제 해결 과정은 학습자 스스로 오류 발견이나 수정 등을 가능하게 하며 학습자의 논리적 능력을 향상시켜준다[2]. 셋째, 프로그래밍은 자신의 생각을 컴퓨터로 구현해 봄으로써 컴퓨터의 동작원리 등과 같은 하드웨어에 대한 지식을 높일 수 있다[11].

이상에서 살펴본 바와 같이, 프로그래밍 학습은 컴퓨터의 동작원리를 이해하는데 많은 도움을 주며 폭넓은 컴퓨터 소양을 길러줄 수 있다[6]. 즉, 프로그래밍 학습은 컴퓨터 과학에서 요구하는 다양한 고등사고능력을 신장시키며 실생활의 문제들을 해결할 수 있는 능력을 향상시킬 수 있다. 이에 따라, 학교 현장에서는 학습자를 대상으로 프로그래밍 학습을 하기 위한 다양한 방법이 제시되고 있다.

2.2 프로그래밍 학습에서의 성찰

컴퓨터를 이용한 문제 해결 과정은 다음과 같이 네 가지로 구분할 수 있다[29]. 첫째, 주어진 문제의 이해 단계, 둘째, 주어진 문제를 해결하기 위해 논리적인 절차인 알고리즘을 작성하는 단계, 셋째, 작성된 알고리즘을 구현하는 단계, 넷째, 유지보수 단계이다. 일반적으로 프로그래밍 교육은 알고리즘을 만드는 것보다는 프로그래밍 언어를 익히는데 치중하였다. 그래서 학습자는 문제를 해결하는 초기 단계인 알고리즘 작성에 어려움을 느끼게 된다[13][14]. 많은 연구들에서도 초보 프로그래머나 학습자가 프로그래밍 과정에서 가장 어려움을 느끼는 부분이 알고리즘을 작성하는 단계임을 보고하였다[22][25]. Polya는 문제 해결을 위한 계획 수립의 단계에서 학습자의 어려움을 줄이기 위해서는 학습자 스스로 충분히 사고할 수 있도록 해야 함을 주장하였다[24]. 즉, 알고리즘 수립 단계에서 교사는 학습자가 생각의 과정을 통해 문제를 해결할 수 있도록 도움을 주어야 할 것으로 보인다. Kolb(1984)도 학습자가 당면한 문제에 대해 새로운 방식으로 생각하고 답을 찾아보는 과정을 성찰로 정의하고, 성찰이 학습자의 경험을 지식으로 변화시킬 수 있는 중요한 기저라고 하였다[18].

학습자가 설계한 알고리즘을 컴퓨터로 구현하는 과정에는 많은 학습이 요구된다. 왜냐하면 학습자가 사고하는 방식과 컴퓨터가 실행되는 과정에 대한 차이를 이해하는데 많은 시간이 필요하기 때문이다

[16]. 또한, 프로그래밍이 어려운 것은 학습자의 학습 스타일이 수평적 접근과 수직적 접근으로 구분되는 것과 무관하지 않다[37]. 즉, 프로그래밍 언어의 사용방법과 규칙을 습득하는 것이 수평적 접근이라면, 기존의 지식이나 학습 경험을 활용하여 복합적으로 문제를 해결하는 것은 수직적 접근이다. 프로그래밍을 하기 위해서는 언어의 문법이나 연산자 등을 기억하는 수평적 접근이 선행되어야 하지만, 프로그래밍 능력을 키우기 위해서는 컴퓨터로 구현하여 적용하는 수직적 학습이 요구되어진다. 알고리즘을 컴퓨터로 구현함에 있어서 학습자에게 요구되는 추상화나 비판적 사고 등과 같은 수직적 접근은 Kolb(1984)가 주장하는 성찰 활동으로 해석될 수 있다. 따라서 성찰은 주어진 문제를 해결하기 위한 알고리즘을 작성하는 단계에서 뿐만 아니라 알고리즘을 컴퓨터로 구현하는 과정에서도 필요한 것으로 보인다.

성찰은 프로그램이 올바르게 구현되었는지 확인하고 잘못된 부분을 스스로 찾아서 수정하는 디버깅 과정에서도 필요하다. 많은 연구들은 디버깅이 초보 학습자에게 어려운 과정이라고 하였다[25][28][34]. 교사가 학습자에게 디버깅하는 방법을 알려주려고 할 때 학습자는 어느 부분이 잘못되었는지 생각하지 않고 맹목적으로 교사의 지시를 받아들이는 모습을 보이기도 한다[28]. 이에 학습자가 디버깅을 할 수 있는 여러 가지 전략이 제시되었는데 그 중의 하나가 생각하는(Thinking) 디버깅이다[34]. 생각하는 디버깅은 학습자가 스스로 문제의 원인을 발견하고 해결해야 하는 것으로, 어느 부분이 잘못되었는지 추측하고 다음에 해야 할 일을 생각하도록 하는 것이다. 즉, 잘못된 부분을 생각하여 발견한 후, 수정하는 디버깅 과정은 곧 성찰의 과정이라 할 수 있다.

이상을 토대로 본 연구는 '성찰이란 주어진 문제 상황에서 학습자가 자신의 학습 내용이나 경험을 토대로, 새로운 개념 형성을 통해 문제를 해결하는 것' 이라고 정의하고 프로그래밍을 통한 문제 해결에서 필요한 것으로 보았다. 따라서 학습자가 주어진 알고리즘을 작성하는 단계나, 알고리즘을 컴퓨터로 구현하는 과정 등에서 성찰을 통해 어떻게 문제를 해결해 가는지 확인해 볼 필요가 있다.

3. 연구방법

프로그래밍 학습에서 활동지를 사용한 성찰 활동을 통해 학습자의 프로그래밍 수행 내용이 어떻게 변해가는지를 알아보고자 하는 본 연구는 다음과 같은 방법으로 진행되었다.

3.1 연구설계

본 연구는 초등학교 3-4학년 학습자 6명을 대상으로 진행되었다. 연구를 진행하기 위한 설계는 다음과 같다. 본 연구에 참여할 6명의 학생들의 학습 수준이나 프로그래밍 수준을 동등하게 하기 위해 다음과 같은 조건의 학생들을 모집하였다.

첫째, 학습자는 모두 프로그래밍 경험이 전혀 없

다. 둘째, 학교에서의 성적은 모두 상위권(상위 30%)이다. 셋째, 문제 해결 성향을 측정하는 PSI 검사[31]에서 유사한 성향을 갖는 학생들이다. PSI 검사지는 문제 해결 행동과 태도에 대한 자신의 인지력을 측정하는 32개의 문항으로 각각 6점 평정 척도로 구성된다. 하위요소는 자신감, 접근회피양식, 자신의 통제로 구성되어 있다. '자신감'은 문제를 해결할 수 있다는 자신감을 측정하는 것으로 11개의 문항들로 구성되어 있다. '접근회피양식'은 문제를 해결해가는 과정에서 얼마나 잘 해결해나가고 하는지를 측정하는 것으로 16개의 문항들로 구성되어 있다. '자신의 통제'는 문제를 해결해 갈 때 정서적으로 얼마나 자신을 잘 통제하는지를 측정하는 것으로 5개의 문항으로 구성되어 있다. 따라서 본 연구에서는 자신감, 접근회피양식, 자신의 통제에서 유사한 성향을 갖는 학습자를 대상으로 연구를 진행하였다.

수업은 [그림 1]과 같이 4단계로 구성되었다. 1단계는 언플러그드 활동을 통해 컴퓨터의 기본원리인 입력과 출력에 대한 학습을 하였다. 2단계는 활동지를 사용한 성찰 활동으로 자신의 생각을 정리할 수 있는 성찰의 시간을 갖는 것이다. 3단계는 교육용 프로그래밍 언어인 스크래치를 사용하여 학습자가 작성한 알고리즘을 컴퓨터로 구현해 보는 것이다. 4단계는 알고리즘대로 프로그램이 구현되었는지 확인해보고, 올바르게 구현되지 않은 경우 성찰의 과정을 통해 오류를 발견하고 수정하는 단계이다.



[그림 1] 연구설계

<표 1>은 15차시 수업의 학습 내용을 나타낸 것이다. 1차시~ 3차시는 1수준으로 기본적인 기능을 익히는 단계이다. 4차시~6차시는 2수준으로 기본적인 기능을 활용하여 기초적인 활동을 하는 단계이다. 3수준에서는 센서를 활용한 활동으로 7차시~9차시에 걸쳐 이루어졌다. 4수준에서는 테이프와 센서를 활용한 고난도 활동으로 10차시~13차시에 걸쳐 진행되었다. 그리고 5수준에서는 3차시에 걸쳐 프로젝트를 수행하였다. 각 수준에서는 '언플러그드 활동'과 '활동지를 사용한 성찰활동', '프로그래밍 활동' 그리고 '성찰을 통한 디버깅' 단계를 수행하였다.

3.2 활동내용과 활동목적

본 연구는 프로그래밍 학습을 4 단계로 구분하여 실시하였다. 각 단계별 활동내용과 활동목적은 <표 2>와 같다.

<표 2> 활동내용과 활동목적

단계	활동내용	활동목적
1단계	언플러그드 활동	컴퓨터의 입력과 출력의 원리 이해
2단계	활동지를 사용한 성찰 활동	주어진 문제를 해결하기 위한 알고리즘 작성
3단계	프로그래밍 활동	알고리즘이 올바르게 작성되었는지 확인
4단계	성찰을 통한 디버깅	잘못된 부분을 찾고 수정

1단계 학습의 목적은 학습자가 언플러그드 활동을 통해 컴퓨터의 입력과 출력의 원리를 이해하는 것이다. 학습자는 컴퓨터로 문제를 해결할 때 입력이 올바르게 않으면 출력 또한 올바르게 못하다는 것을 학습한다. 2단계 학습의 목적은 학습자가 활동지를 사용한 성찰 활동을 통해 주어진 문제를 해결하기 위한 알고리즘을 작성하는 것이다. 3단계 학습의 목적은 학습자가 프로그래밍 활동을 통해 알고리즘이 올바르게 작성되었는지 확인하는 것이다. 그리고 4단계 학습의 목적은 학습자가 성찰을 통해 어느 부분이 잘못되었는지 스스로 발견하고 오류를 수정할 수 있도록 하는 것이다.

다음은 각 단계별 구체적인 활동내용에 관한 설명이다.

3.2.1 1단계 - 언플러그드 활동

1단계 언플러그드 활동 수업은 ‘설명하고 그림 그리기’와 ‘숫자로 그림 그리기’로 구분할 수 있다. 각 활동 내용과 예시는 <표 3>과 같다.

첫째, ‘설명하고 그림 그리기’ 활동이다. 본 활동에서는 한 명의 학습자가 앞으로 나와 교사가 준비한 그림에 대해 설명을 한다. 그리고 나머지 학습자는 설명에 따라 그림을 그린다. 교사는 다음과 같은 내용을 설명하였다.

- i) 설명하는 학습자 : 컴퓨터에 명령하는 역할
- ii) 그림 그리는 학습자 : 컴퓨터 역할

컴퓨터 역할을 하는 학습자는 어떠한 질문도 할 수 없다. 활동이 끝난 이후에는 각자의 그림과 정답의 그림이 왜 다른지를 생각해 보도록 하였다.

둘째, ‘숫자로 그림 그리기’ 활동이다. 본 활동은 두 명씩 짝을 지어서 한 명은 숫자를 말하고 다른 명은 정해진 규칙에 따라 그림을 그리는 활동이다. 첫 번째 활동과 마찬가지로 설명을 듣는 학습자

<표 1> 프로그래밍 학습 내용

수준	차시	학습 내용		활동 단계
1수준 (기능 익히기)	1차시	숫자를 이용하여 그림 그리기		언플러그드 활동
	2차시	1단계 : 로봇을 움직이는 명령어 익히기 2단계 : 장애물을 피해서 로봇 이동시키기		활동지를 사용한 성찰활동
	3차시	스크래치로 구현하기(1-2단계)		프로그래밍 활동 성찰을 통한 디버깅
2수준 (기초적 활동)	4차시	그림을 그리고 숫자로 표현하기		언플러그드 활동
	5차시	3단계 : 선물가게에서 영희의 선물 가져오기 4단계 : 선물을 푸는데 필요한 열쇠 가져오기 5단계 : 영희에게 선물 가져가기 6단계 : 영희에게 열쇠 가져가기		활동지를 사용한 성찰활동
	6차시	스크래치로 구현하기(3-6단계)		프로그래밍 활동 성찰을 통한 디버깅
3수준 (센서를 활용한 난이도 상황)	7차시	컴퓨터와 프로그래머		언플러그드 활동
	8차시	센서 이용하기	7단계 : 로봇 움직이기 8단계 : 장애물 피하기 9단계 : 선물 가져오기 10단계 : 열쇠 가져오기 11단계 : 영희에게 선물주기 12단계 : 영희에게 열쇠주기	활동지를 사용한 성찰활동
4수준 (테이프와 센서를 활용한 고난도 활동)	9차시	스크래치로 구현하기(7-12단계)		프로그래밍 활동 성찰을 통한 디버깅
	10차시	테이프 이용하기	13단계 : 선물 가져오기 14단계 : 열쇠 가져오기 15단계 : 영희에게 돌아오기	활동지를 사용한 성찰활동
		두 개의 센서 이용하기	16단계 : 로봇 움직이기 17단계 : 선물과 열쇠 가져오기 18단계 : 영희에게 돌아오기	
	11차시	스크래치로 구현하기(13-18단계)		프로그래밍 활동 성찰을 통한 디버깅
5수준 (프로젝트 수행)	12차시			
	13차시	로봇 문제 설계하기		활동지를 사용한 성찰활동
	14차시	스크래치로 구현하기		프로그래밍 활동 성찰을 통한 디버깅
	15차시	프로젝트 발표하기		

는 질문을 할 수 없다. 활동이 끝난 이후, 학습자는 정답과 자신의 그림을 비교하고 다른 이유에 대해 토의하였다.

<표 3> 1단계 - 언플러그드 활동

활동 이름	활동내용	활동예시
설명하고 그림 그리기	교사 - 그림 준비	
	학습자 I - 준비한 그림 설명 나머지 학습자 - 설명 듣고 그림 그리기	
	정답과 차이 비교	
숫자로 그림 그리기	교사 - 활동지 준비	★ 숫자로 그림 그리기
	학습자 I - 활동지에 적힌 숫자 말하기	
	학습자 II - 숫자를 이용한 그림 그리기	
정답과 차이 비교		

3.2.2 2단계 - 활동지를 사용한 성찰 활동

2단계 활동지를 사용한 성찰 활동은 ‘분석’, ‘설계’, ‘구현’, ‘수정’ 단계로 진행한다. 학습자는 제시된 활동을 수행하기 위해 목적지까지 이르기 위한 알고리즘을 작성하고, 스스로의 활동에 대해 평가하고, 수정하여 활동지를 완성하였다. 활동의 내용과 예시는 <표 4>와 같다.

<표 4> 2단계 - 활동지를 사용한 성찰 활동

단계	활동내용	활동예시
분석	출발과 도착위치 확인	
설계	활동지를 사용한 성찰 활동에서 사용되는 명령어	()까지 가기 (오른쪽)으로 90° 돌기 (왼쪽)으로 90° 돌기 ()회 반복 ()달으면
구현	교사에게 정답확인 및 교사의 정답, 오답 반응	2번째 명령어에서 오른쪽이 아닌 채점해주시기 아래 라고 하셨기 때문에 (시점이달려서)

수정	알고리즘 수정	명령어 적어보기(1차)	명령어 적어보기(2차)
		2번째 명령어에서 오른쪽이 아닌 채점해주시기 아래 라고 하셨기 때문에	1이후 1까지 정진 1이후 2까지 정진 1이후 3까지 정진 1이후 4까지 정진 1이후 5까지 정진 1이후 6까지 정진 1이후 7까지 정진 1이후 8까지 정진 1이후 9까지 정진 1이후 10까지 정진 1이후 11까지 정진 1이후 12까지 정진 1이후 13까지 정진 1이후 14까지 정진 1이후 15까지 정진 1이후 16까지 정진 1이후 17까지 정진 1이후 18까지 정진 1이후 19까지 정진 1이후 20까지 정진 1이후 21까지 정진 1이후 22까지 정진 1이후 23까지 정진 1이후 24까지 정진 1이후 25까지 정진 1이후 26까지 정진 1이후 27까지 정진 1이후 28까지 정진 1이후 29까지 정진 1이후 30까지 정진 1이후 31까지 정진 1이후 32까지 정진 1이후 33까지 정진 1이후 34까지 정진 1이후 35까지 정진 1이후 36까지 정진 1이후 37까지 정진 1이후 38까지 정진 1이후 39까지 정진 1이후 40까지 정진 1이후 41까지 정진 1이후 42까지 정진 1이후 43까지 정진 1이후 44까지 정진 1이후 45까지 정진 1이후 46까지 정진 1이후 47까지 정진 1이후 48까지 정진 1이후 49까지 정진 1이후 50까지 정진 1이후 51까지 정진 1이후 52까지 정진 1이후 53까지 정진 1이후 54까지 정진 1이후 55까지 정진 1이후 56까지 정진 1이후 57까지 정진 1이후 58까지 정진 1이후 59까지 정진 1이후 60까지 정진 1이후 61까지 정진 1이후 62까지 정진 1이후 63까지 정진 1이후 64까지 정진 1이후 65까지 정진 1이후 66까지 정진 1이후 67까지 정진 1이후 68까지 정진 1이후 69까지 정진 1이후 70까지 정진 1이후 71까지 정진 1이후 72까지 정진 1이후 73까지 정진 1이후 74까지 정진 1이후 75까지 정진 1이후 76까지 정진 1이후 77까지 정진 1이후 78까지 정진 1이후 79까지 정진 1이후 80까지 정진 1이후 81까지 정진 1이후 82까지 정진 1이후 83까지 정진 1이후 84까지 정진 1이후 85까지 정진 1이후 86까지 정진 1이후 87까지 정진 1이후 88까지 정진 1이후 89까지 정진 1이후 90까지 정진 1이후 91까지 정진 1이후 92까지 정진 1이후 93까지 정진 1이후 94까지 정진 1이후 95까지 정진 1이후 96까지 정진 1이후 97까지 정진 1이후 98까지 정진 1이후 99까지 정진 1이후 100까지 정진

첫째, 분석 단계에서 학습자는 출발과 도착 위치를 확인한다. 교사는 학습자에게 문제를 해결하기 위해 고려해야 할 사항인 출발지점에서 도착지점까지의 경로를 펜으로 그려보도록 한다. 둘째, 설계 단계에서 문제를 해결하는데 필요한 명령어를 제시하여 학습자가 문제 해결에 집중할 수 있도록 하였다. 따라서 학습자는 주어진 명령어를 사용하여 문제 해결 과정을 작성한다. 셋째, 구현 단계에서 학습자는 자신이 작성한 내용을 교사에게 확인받는다. 이때, 교사는 컴퓨터의 역할이기 때문에, 정답이나 오답 여부만 알려줄 뿐 수정해야 할 내용을 알려주지는 않는다. 즉, 학습자 스스로 생각하고 잘못된 부분을 수정하는 성찰 활동을 독려한다. 넷째, 수정 단계에서는 학습자가 알고리즘을 올바르게 수정할 때까지 ‘구현’ 단계를 반복한다.

3.2.3 3단계 - 프로그래밍 활동

3단계에서는 자신이 작성한 알고리즘을 바탕으로 교육용 프로그래밍 언어인 스크래치를 이용하여 컴퓨터로 구현해보도록 하였다. 학습자가 활동지를 통해 문제를 해결했던 내용을 스크래치를 통해 구현하도록 하였다. 학습자는 스크래치를 처음 접했기 때문에 문제를 해결하기 위해 필요한 기본적인 명령어(전진, 좌회전, 우회전, 센서 전진 등)에 대해 숙지하였다. <표 5>에서 보는 바와 같이, 활동지를 통해 문제를 해결했을 때의 명령어와 스크래치 명령어는 매우 유사한 형태로 제시되었다.

<표 5> 성찰 활동 명령어와 스크래치 명령어

성찰 활동 명령어	스크래치 명령어	설명
()까지 가기	10 만큼 움직이기	어떤 개체에 닿을 때까지 전진
(오른쪽)으로 90° 돌기	90도 우회전	90도 우회전
(왼쪽)으로 90° 돌기	90도 좌회전	90도 좌회전
()회 반복	반복 10 회	반복 횟수 지정
()달으면	만약 색에 닿기? 리면	어떤 색에 닿았는지 확인

3.2.4 4단계 - 성찰을 통한 디버깅

4단계에서는 학습자가 스크래치로 구현하는 과정에서 올바른 결과를 위해 스스로 성찰할 수 있도록 하였다. 활동지의 내용이 스크래치로 구현되지 않았을 때, 교사는 학습자에게 자신이 작성한 알고리즘과 비교해보도록 하여 스스로 오류를 찾고 문제를 해결할 수 있도록 지도하였다.

4. 연구결과

본 연구의 결과는 연구 실행 단계에 따라 프로그래밍 활동에서 학습자가 어떻게 문제를 해결했는지 에 대한 것이다. 각 단계별 활동 내용과 학습자들의 목표 달성 정도에 대해 살펴보면 다음과 같다.

4.1 1단계 - 언플러그드 활동 결과

학습자는 1단계 언플러그드 활동을 통해 컴퓨터의 입력과 출력의 원리를 학습하였다. <표 6>은 언플러그드 활동 중 '설명하고 그림 그리기' 활동의 결과를 나타낸 것이다.

<표 6> '설명하고 그림 그리기' 활동 결과

설명한 그림	학습자 I	학습자 II

<표 6>에 나타난 바와 같이 '설명하고 그림 그리기' 활동의 결과를 살펴보면 두 명의 학습자는 같은 설명을 들었지만 서로 다른 그림을 그린 것을 확인할 수 있다. 즉, 학습자는 설명하는 사람이 일정한 규칙이 없을 경우, 설명을 듣는 사람이 각자 다르게 받아들일 수 있으며, 다른 결과를 얻게 된다는 것을 인식한다. 이 활동에서 학습자는 컴퓨터가 정확하게 실행할 수 있는 형태로 컴퓨터에 명령을 내려야 하는 것에 대해 알게 되었다.

<표 7>은 '입력과 출력'에 대한 학습으로 '숫자로 그림그리기' 활동의 결과와 산출물에 대한 학습자의 대화를 나타낸 것이다.

<표 7>에 나타난 바와 같이 학습자는 입력이 잘못되면, 출력이 잘못될 수 있다는 것을 스스로 인식하게 되었다.

4.2 2단계 - 활동지를 사용한 성찰 활동 결과

2단계에서 학습자는 활동지를 사용해 문제 해결 과정을 기록하고, 지속적인 성찰을 통해 문제 해결을 달성할 수 있도록 연습하였다. <표 8>은 주어진 문제를 해결하기 위해 학생이 작성한 알고리즘을 나타낸 것이다.

활동지 학습에서 교사는 알고리즘의 정확성을 판단하기 위해 컴퓨터 역할을 하였다. 즉, 교사는 학습자가 작성한 문제 해결 과정을 따라 실행하고 알

<표 7> '숫자로 그림 그리기' 활동 결과

활동 결과		
활동2 : 숫자로 그림 그리기		
	4.11	5
	4.9	2.1
	4.9	2.1
	4.11	
	4.9	5
	4.9	
	5.7	
	0.17	
	1.15	

학습자1 : "정답은 무슨 그림일까?"
학습자2 : "컵 인 것 같은데?"
학습자1 : "그런데 손잡이 부분이 이상한데? 어디 잘못된 것 아니야?"
학습자2 : "내가 불러준 대로 색칠했을 뿐인데?"
학습자1 : "두 번째와 세 번째 줄이 4-9-2-1 맞아?"
학습자2 : "4-9-1-2로 불러준 것 같은데?"
학습자1 : "아! 내가 잘못 알려주었구나"
학습자2 : "네가 잘못 알려주면 내가 올바르게 그릴 수가 있지"

<표 8> 활동지를 사용한 성찰 활동 결과

활동 결과		
명령어 적어보기(1차)	명령어 적어보기(2차)	명령어 적어보기(3차)
기름 칠해서 까맣게	흰색에 노란색 칠하기	검은색 칠하기
검은색 칠하기	검은색 칠하기	검은색 칠하기
노란색 칠하기	노란색 칠하기	노란색 칠하기
노란색 칠하기	노란색 칠하기	노란색 칠하기
노란색 칠하기	노란색 칠하기	노란색 칠하기
노란색 칠하기	노란색 칠하기	노란색 칠하기
노란색 칠하기	노란색 칠하기	노란색 칠하기
노란색 칠하기	노란색 칠하기	노란색 칠하기

고리즘의 정확성 여부에 대해서만 판단하였다. 문제 해결이 올바르지 않다면, 교사는 학습자로 하여금 스스로 올바르지 않은 부분을 찾고 해결하는 성찰의 시간을 갖도록 도와주었다. 보통 10분 정도의 시간이 소요되었고, 15 - 20분 정도의 시간이 소요된 2명의 학습자에게 교사는 작성된 알고리즘을 따라 실행하도록 하여 학습자 스스로 잘못된 부분을 찾을 수 있도록 독려했다.

다음 대화는 학습자가 본인의 알고리즘에서 잘못된 부분을 스스로 찾을 수 있도록 하기 위해 교사가 학습자에게 성찰을 독려하는 것을 나타낸 것이다.

교사 : "로봇이 도착점까지 가지 않네요?"
학습자 : "음.. 아무리 봐도 어느 부분이 잘못되었는지 모르겠어요"
교사 : "작성한 대로 선생님이 따라가 볼게요. 검은색까지 움직인 다음에 왼쪽으로 회전해요."
학습자 : "아! 왼쪽으로 회전해야 해요"
교사 : "뒷부분도 잘못된 부분을 찾고 수정할 수 있나요?"
학습자 : "네, 명령어를 하나씩 수행해보면 어느 부분이 잘못되었는지 찾을 수 있을 것 같아요"

학습자는 컴퓨터를 사용하여 결과를 확인하지 않

고 성찰을 통해 알고리즘의 정확성을 판단할 수 있었다. 처음에는 자신이 작성한 알고리즘의 어느 부분이 잘못되었는지 발견하지 못했지만, 교사의 도움으로 한 번 더 생각하고, 주의 깊은 사고를 통해 잘못된 부분을 스스로 찾고 수정해서 알고리즘을 완성하였다.

이와 같이 본 연구를 진행하면서, 학습자가 문제를 해결하는 과정에서 어려움을 느낄 때마다 교사의 도움을 받아 성찰 활동을 하거나 문제를 해결한 평균 횟수는 <표 9>와 같다.

<표 9> 문제 해결 과정에서 교사의 도움을 받은 평균 횟수

A 학습자	B 학습자	C 학습자	D 학습자	E 학습자	F 학습자
1	2	3	3	3	5

문제 해결 과정에서 A, B 학습자는 자신이 작성한 알고리즘을 교사에게 확인 받기 전에 스스로 정확성을 판단하는 모습을 보였다. 따라서 교사에게 질문하거나 요청하는 횟수가 상대적으로 적음을 알 수 있다. 그러나 C, D, E, F 학습자는 문제 해결이 어려울 때마다 즉각적으로 교사의 도움을 요청하였다. 즉, 스스로의 판단보다는 교사의 도움에 의존하는 모습을 보였다. A, B 학습자는 교사가 지시한 대로 성찰에 몰입하는 모습을 보였기 때문에 교사는 학습자가 성찰의 시간을 충분히 가질 수 있도록 독려했다. 이에 반해 C, D, E, F 학습자는 교사가 성찰을 통해 문제를 해결할 것을 독려했음에도 불구하고 스스로 문제를 풀기 위한 성찰에 몰입을 하지 않고 교사의 도움을 받아서 즉각적으로 해결하려고 하였다. 이처럼 문제 해결 성향이 유사한 학습자에게 동일한 성찰을 유도했음에도 불구하고 교사의 도움을 받은 횟수가 달라지는 이유는 PSI가 자신감, 접근회피양식, 자신의 통제와 같이 도전적인 측면에 대한 것일 뿐, 문제를 해결하기 위해 스스로 생각하고 조직하는 것에 대한 문제 풀이 방법은 아닌 것으로 해석할 수 있다. 즉, 문제를 푸는데 있어서 동기가 유사하다고 할지라도 푸는 방법은 다를 수 있기 때문이다.

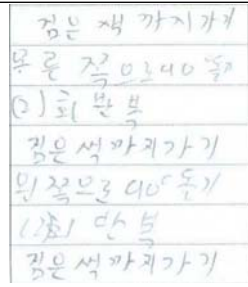

본 연구 결과는 순서도를 이용하여 자신만의 문제 해결 방법을 만들어낸 학습자가 교사의 개입에 따라 문제 해결을 한 학습자에 비해 창의적으로 문제를 해결하며 복잡한 문제를 쉽게 해결한다는 연구와 일치한다[35]. 따라서 학습자가 프로그래밍 과정에서 문제를 보다 잘 해결하도록 하기 위해서는 학습자 스스로 생각하고, 문제를 해결할 수 있는 기회를 제공해 줄 필요가 있다.

4.3 3단계 - 프로그래밍 활동 결과

3단계에서 학습자는 교육용 프로그래밍 언어인 스크래치를 이용하여 알고리즘을 구현하였다. 프로그래밍 활동에서 학습자는 자신이 작성한 알고리즘보다 스크래치에서 제공하는 기능을 이용한 문제 해결에 더 많은 관심을 보였다. 따라서 교사는 학습자

들이 문제 해결에 집중할 수 있도록 하기 위하여 알고리즘을 바탕으로 프로그래밍 활동을 하도록 지도하였다. 다음 <표 10>은 학습자가 작성한 알고리즘에 따라 스크래치를 사용하여 구현한 결과를 나타낸 것이다.

<표 10> 교육용 프로그래밍 언어를 사용한 구현 활동 결과

알고리즘	구현결과
	

본 연구에 참여한 학습자 중 알고리즘 작성단계에서 교사의 도움을 많이 받았던 C, D, E, F 학습자는 스크래치로 구현하는 과정에서도 알고리즘을 보고, 생각해서 블록을 선택하기보다 시행착오를 거쳐 문제를 해결하는 모습을 보였다. 문제 해결에 적합한 블록을 선택하지 않고, 명령어를 둘러보고 결과를 확인하는 시행착오의 시간이 길어지면서, 구현을 어려워하는 모습을 보였다. 그리고 C, D, E, F 학습자의 시행착오를 통해 문제를 해결한 학습자는 문제를 해결한 이후에도 어떻게 문제를 해결했는지 설명하지 못한다는 연구와 일치된 결과를 나타내었다[35].

이와 달리 교사의 도움 없이 스스로 문제 해결 과정을 작성한 학습자는 주어진 문제를 보고 자신이 작성한 알고리즘에 따라 스크래치 명령어를 선택하는 모습을 보였다. 이는 문제 해결 과정을 적는 것이 프로그래밍에 도움을 주며 체계적으로 문제를 해결하도록 한다는 연구 결과와 일치함을 알 수 있다[35]. 또한 문제 해결 과정을 적는 것은 프로그래밍의 계획단계이며 이 과정은 프로그래밍에서 반드시 필요한 단계임을 알 수 있다[27][32].

4.4 4단계 - 성찰을 통한 디버깅 결과

4단계에서는 알고리즘을 바탕으로 컴퓨터로 구현할 때 문제가 올바르게 해결되지 않는 경우 알고리즘과 스크래치를 비교해보도록 하여 학습자 스스로 문제를 해결할 수 있도록 하였다. 다음은 알고리즘을 올바르게 작성하였지만 스크래치로 구현하는 과정에서 어려움을 겪고 있는 학습자가 어느 부분이 잘못되었는지 스스로 발견하는 과정을 나타낸 것이다.

학습자 : “선생님, 로봇이 이상한 곳으로 움직여요”
 교사 : “로봇이 어느 부분까지 올바르게 움직이고 있나요?”
 학습자 : “(스크래치 화면을 가리키며) 이 부분이에요”
 교사 : “알고리즘에서는 어느 부분까지 올바르게 움직이고 있나요?”
 학습자 : “(자신이 작성한 알고리즘을 가리키며) 이 부분이에요”
 교사 : “그러면, 다음 알고리즘과 스크래치로 구현한 부분을 비교해 볼까요? 같은가요?”
 학습자 : “아! 이 부분이 잘못 되었어요”

위의 대화에서 확인할 수 있듯이 학습자는 디버깅 과정에서 어느 부분이 잘못되었는지 스스로 발견하는 것을 알 수 있다. 그리고 교사의 제안에 따라 자신이 작성한 알고리즘과 스크래치 명령어를 비교하여 디버깅 활동을 했던, A와 B 학습자는 다른 학습자에 비해 평균적으로 7~8분 정도 빠른 시간 안에 문제를 해결하는 모습을 보였다. 본 연구의 결과는 프로그래밍 과정에서 디버깅이나 에러를 수정하기 위해서는 성찰의 자세가 필요하다는 연구와 일치한다[17]. 프로그래밍 과정은 하나의 완성된 문제 해결을 나타내는 것이기 때문에 계획, 설계, 구현 등의 단계가 일관성을 지녀야 한다. 따라서 문제를 해결하기 위한 성찰은 프로그래밍의 특정 부분에서가 아니라 모든 부분에서 일관성 있는 사고를 가능하게 할 수 있을 것이다.

4.5 프로젝트 수행 결과

학습한 내용을 바탕으로 학습자는 자신이 원하는 프로젝트를 완성하기 위한 과제를 부여받았다. 프로젝트 수행은 계획, 알고리즘 작성, 그리고 구현의 단계로 이루어진다.

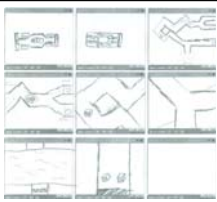

- i) 프로젝트 계획하기
- ii) 알고리즘 작성하기
- iii) 스크래치로 구현하기

첫째, 학습자는 자신들이 구현하고자 하는 프로젝트를 계획하였다. 이 과정에서 스스로 성찰을 통해 문제를 해결했던 A 학습자는 다른 학습자보다 복잡하고 다양한 형태의 프로젝트를 계획함을 알 수 있었다. <표 11>은 학습자의 프로젝트 계획 화면을 나타낸 것이다.

둘째, 알고리즘을 작성하는 과정에서 A 학습자는 기존에 학습한 내용을 바탕으로 자신만의 문제 해결 과정을 작성하였다. 이에 반해 E 학습자는 기존의 수업과 비슷한 형태의 문제 해결 과정을 작성하였다.

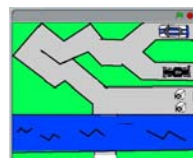


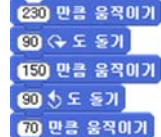



셋째, 구현하는 과정에서 A 학습자는 알고리즘대로 스크래치 명령어를 나열하였으며 기존에 학습한 스크래치 기능 이외에 새로운 명령어를 사용하는 모습을 보였다. E 학습자는 자신이 작성한 알고리즘과는 달리 스크래치 명령어 사용에서 시행착오를 통해 문제를 해결하였다.

<표 11> 성찰 활동에 따른 개인 프로젝트 화면 계획 결과

구분	학습자 A	학습자 E
프로젝트 화면 계획		

<표 12>는 프로젝트 계획에 따라 학습자가 완성한 최종 프로젝트 화면과 사용한 명령어를 나타낸 것이다.

<표 12> 성찰 활동 유무에 따른 개인 프로젝트 결과 화면

구분	A 학습자	E 학습자
프로젝트 화면		
개체 1		
스크래치 명령어		사용하지 않음
개체 2		
개체 3		

프로젝트 결과를 통해 A 학습자는 다양하고 복잡한 형태의 개체 및 스크래치 명령어를 사용한 것을 확인할 수 있다. E 학습자는 기존에 학습한 내용과 비슷한 형태의 맵과 명령어를 사용한 것을 확인할 수 있다. 이는 문제 해결 과정을 적고, 자신의 생각을 잘 정리할 수 있도록 가르치는 것이 학습자가 창의적으로 문제를 해결하는데 도움을 준다는 연구 결과와 일치함을 알 수 있다[35].

5. 결론

본 연구는 주어진 문제를 해결하기 위해 문제풀이 과정에 대해 생각하고, 절차를 적는 행위가 프로젝트 수행에 어떤 도움을 주는지 알아보기 위한 목적이 있었다. 연구의 목적을 달성하기 위해 프로그래밍 과정에서 학습자의 문제 해결과정을 분석하였다. 분석을 위해 본 연구는 초등학교 3-4학년 6명의 학습자를 대상으로 활동지를 사용한 프로그래밍 수업을 15차시에 걸쳐 진행하였다.

연구결과, 성찰을 통해 알고리즘을 작성한 학습자는 자신의 생각을 논리적으로 정리하여 프로그래밍을 수행하였다. 또한 프로젝트를 진행하는 과정에서도 다양하고 창의적인 프로젝트를 완성하였다. 이에 반해 성찰의 과정없이 문제를 해결한 학습자는 단순한 알고리즘 작성과 시행착오를 통해 프로젝트를 완성하였다. 따라서 성찰의 시간을 거쳐 절차적으로 문제 해결 과정을 적는 것이 프로젝트 수행에 도움을 주었던 것으로 해석할 수 있다. 이러한 결과는 문제 해결 절차를 적는 과정이 프로그래밍에 대한 학습자들의 이해를 높이며 체계적으로 문제를 해결하는데 도움을 주는 생각하는 활동이라는 연구와 일치하는 결과이다[21][33][35][36].

본 연구를 토대로 할 때, 초등학생의 프로그래밍 학습도 시행착오를 통해 구현된 것 보다는 문제 풀이 과정에 대해 생각하고 활동지를 통해 자신의 생각을 정리하는 성찰 중심의 프로그래밍 학습이 이루어져야 함을 알 수 있었다. 즉, 프로그래밍 학습에서 문제 해결의 목적을 잃지 않고 일관성 있고 체계적으로 생각할 수 있도록 성찰의 과정을 독려할 필요가 있다. 그리고 학습자가 작성한 문제 해결 과정을 컴퓨터로 구현할 때 발생하는 오류에 대해 즉각적인 도움을 주기 보다는 성찰을 통해 스스로 문제를 발견하고 수정하도록 하여 문제 해결력을 높일 수 있도록 해야 할 것이다.

참고문헌

[1] 교육인적자원부 (2007), 교육인적자원부 고시 제 2007-79호[별책3] 중학교 교육과정, 서울:교육인적자원부.
 [2] 김갑수 (2010), 초등학생들의 창의력과 논리력 향상을 위한 프로그래밍 언어 교수전략에 관한 연구, 정보교육학회 논문지, 14-1, 89-97.
 [3] 김정훈 · 이원규 · 김성식 · 강신천 · 강의성 · 김영식 · 유현창 (2007), 중학교 교과재량활동 I (한문, 정보, 환경) 교육과정해설 연구개발. 연구보고 CRC 2007-24, 한국교육과정평가원.
 [4] 김수환, 이원규, 김현철 (2009), 개정된 정보교육 과정에서 교육용프로그래밍언어의 교육적 적용 방안, 컴퓨터교육학회 논문지, 12-2, 23-31.
 [5] 김수환, 한선관, 김현철 (2010), Computational Literacy 교육에서 프로그래밍 능력과 학습자 특성에 관한 연구, 컴퓨터교육학회 논문지, 13-2,

15-23.
 [6] 김종한, 최현중, 김태영 (2011), 선행조직자를 적용한 프로그래밍 학습이 초등학생의 논리적 사고력과 자기효능감에 미치는 영향, 정보교육학회 논문지, 15-2, 189-199.
 [7] 김종해, 김정훈, 이원규 (2008), 2007년 개정 중학교 정보 교육과정의 “문제 해결 방법과 절차” 영역 성취기준 및 평가기준 개발 방안 연구, 컴퓨터교육학회 논문지, 11-6, 39-51.
 [8] 문외식 (2007), 교육용로봇을 이용한 프로그래밍 학습 모형, 정보교육학회 논문지, 11-2, 231-241.
 [9] 배학진, 이은경, 이영준 (2009), 문제 중심 학습을 적용한 스크래치 프로그래밍 교수 학습 모형, 컴퓨터교육학회 논문지, 12-3, 11-22.
 [10] 백영균, 우인상 (1994), LOGO 프로그래밍의 수업방법이 문제해결력에 미치는 효과에 관한 연구, 교육공학회 논문지, 9-1, 73-90.
 [11] 유정수, 이민희 (2009), 두리틀을 이용한 프로그래밍 수업이 창의성, 문제해결력, 프로그래밍 흥미도 향상에 미치는 영향, 정보교육학회 논문지, 13-4, 443-450.
 [12] 장윤재, 김자미, 이원규 (2011), 교육용 프로그래밍 언어를 활용한 학습에서 학습양식에 따른 프로젝트 완성 능력, 컴퓨터교육학회 논문지, 14-1, 1-12.
 [13] 최현중 (2011), 대학 프로그래밍 강좌를 위한 프로그래밍 교육 프레임워크, 컴퓨터교육학회 논문지, 14-1, 69-79.
 [14] A. Gomes, A. J. Mendes (2007), Learning to Program - Difficulties and Solutions, in International Conference on Engineering Education - ICEE 2007, 283-287.
 [15] A. Tucker, F. Deek, J. Jones, D. McCowan, C. Stephenson, A. Verno (2003), A model curriculum for K-12 computer science, Final report of the ACM K-12 task force curriculum committee.
 [16] B. D. Boulay (1989), Some difficulties of learning to program, in Studying the Novice Programmer, 283-299.
 [17] C. Bravo, M.J. Marcelino, A. Gomes, M. Esteves, A.J. Mendes (2005), Integrating Educational Tools for Collaborative Computer Programming Learning, Journal of Universal Computer Science, 11-9, 1505-1517.
 [18] D. A. Kolb (1984), Experiential learning, Englewood Cliffs, NJ: Prentice-Hall.
 [19] D. Gupta (2004), What is a good first programming language?, Crossroads: The ACM Student Magazine, 10-4, 7.
 [20] E. Milková, M. Turčáni (2006), Digital objects supporting development of algorithmic thinking, Current Developments in

- Technology-Assisted Education, 376-380.
- [21] F. Deek, S. R. Hiltz, H. Kimmel, N. Rotter (1999), Cognitive assessment of students' problem solving and program development skills, *Journal of Engineering Education*, 88-3, 317 - 326.
- [22] F. Détienne (1990), Expert programming knowledge: A schema based approach, In J.-M. Hoc, T.R.G. Green, D.J. Gilmore and R. Samurçay (Eds.), *Psychology of Programming*, 205-222, London: Academic Press.
- [23] G. Futschek (2006), Algorithmic Thinking: The Key for Understanding Computer Science, In *Lecture Notes in Computer Science 4226*, Springer, 159-168.
- [24] G. Polya (1957), *How To Solve It*, Garden City, NJ: Doubleday.
- [25] J. C. Spohrer, E. Soloway (1989), Novice mistakes: Are the folk wisdoms correct?, *Communications of the ACM*, 624-632.
- [26] J. Maloney, K. Pepler, Y. Kafai, M. Resnick, N. Rusk (2008), Programming by choice: Urban youth learning programming with Scratch, *Proceedings of 39th SIGCSE Technical Symposium on Computer Science Education*, 40-1.
- [27] K. Robinson (1998), *All our Futures: Creativity, Culture and Education*, Report to National Advisory Committee on Creative and Cultural Education, <http://www.sirkenrobinson.com/skr/pdf/allourfutures.pdf> accessed 31 October 2011.
- [28] L. MURPHY, G. LEWANDOWSKI, R. MCCAULEY, B. SIMON, L. THOMAS, C. ZANDER (2008), Debugging: the good, the bad, and the quirky -- a qualitative analysis of novices' strategies, In *SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer science education* (New York, NY, USA, 2008), ACM, 163-167.
- [29] N. Dale, J. Lewis (2009), computer science illuminated.
- [30] P. A. G. Sivilotti , S. A. Laugel (2008), Scratching the surface of advanced topics in software engineering: a workshop module for middle school students, *Proceedings of 39th SIGCSE Technical Symposium on Computer Science Education*, 40-1.
- [31] P. P. Heppner, T. E. Witty, W. A. Dixon (2004), Problem-solving appraisal and human adjustment: A review of 20 years of research utilizing the problem solving inventory [Major contribution], *The Counseling Psychologist*, 32, 344-428.
- [32] Queensland Studies Authority (2004), *Technology: Years 1 - 10 syllabus*, Brisbane, Queensland: The State of Queensland, http://www.qsa.qld.edu.au/yrs1_10/kla/technology/index.html accessed 10 February 2004.
- [33] R. Satchwell (1997), Using functional flow diagrams to enhance technical systems understanding, *Journal of Industrial Teacher Education*, 34-2, 1-28, <http://scholar.lib.vt.edu/ejournals/JITE/v34n2/Satchwell.html> accessed 31 October 2011.
- [34] S. Fitzgerald, R. McCauley, B. Hanks, L. Murphy, B. Simon, C. Zander (2010), Debugging From the Student Perspective, *Education, IEEE Transactions on*, 53-3, 390-396.
- [35] S. J. Norton, C. J. McRobbie, I. S. Ginns (2007), problem solving in a middle school robotics design classroom, *Research in Science Education*, 37-3, 261-277.
- [36] T. Crew, J. Butterfield (2003), Improving the learning environment in beginning programming classes: An experiment in gender equity, *Journal of Information Systems Education*, 14-1, 69 - 76.
- [37] T. Jenkins (2002), On the Difficulty of Learning to Program, *Proceedings of 3rd LTSN-ICS Conference*, 53-58.

저자소개



김 용 천

2010년 2월: 고려대학교 컴퓨터교육과 (이학사)
2010년~현재 : 고려대학교 대학원 컴퓨터교육학과 석사과정

관심분야 : 컴퓨터교육, 프로그래밍교육, 알고리즘
e-mail : yongcheon.kim@inc.korea.ac.kr



김 자 미

1992년 8월: 이화여자대학교 교육학과 (문학사)
1995년 2월: 이화여자대학교 교육학과 (문학석사)
2011년 8월 : 고려대학교 대학원 컴퓨터교육학과 (이학박사)
2011년~현재 : 고려대학교 연구정보분석센터 연구교수

관심분야 : 컴퓨터교육, 교육정보화평가, 이러닝
e-mail : jamee.kim@inc.korea.ac.kr



이 원 규

1985년 2월: 고려대학교 문과대학 영어영문학과 (문학사)
1989년 2월: 筑波大學 大學院 理工學研究科 (공학석사)
1993년 2월: 筑波大學 大學院 工學研究科 (공학박사)

1993년 3월 : 한국문화예술진흥원 책임연구원
1996년~현재 : 고려대학교 사범대학 컴퓨터교육과 교수

관심분야 : 컴퓨터교육, 정보검색, 데이터베이스
e-mail : lee@inc.korea.ac.kr