

---

# 'Dolittle'을 활용한 프로그래밍 수업이 논리적 사고에 미치는 효과

권창미\* · 권보섭\*\*

The Effects of Programming Lessons using 'Dolittle' on Logical Thinking

Chang-Mi Kwon\* · Boseob Kwon\*\*

## 요 약

지식정보사회에서 살아갈 사람들에게 중요시되는 능력은 새로운 상황에 당면했을 때, 새로운 아이디어를 개발하고 이를 적용해 주어진 문제를 해결할 수 있는 논리적 사고력이 요구 된다. ACM은 논리적 사고력 향상을 위해 프로그래밍언어를 통한 교육이 필요하다고 제시하고 있으나 현재 사용되고 있는 교육용 프로그래밍언어는 개념 중심이며 차후 학습으로의 연계성이 상당히 부족하다. 이에 이 논문에서는 보다 효과적인 논리적 사고력 형성을 위해 교육용프로그래밍 언어인 '두리틀'을 소개하고 이를 실제 수업에 적용하여, 인지발달 수준과 논리적 사고력 형성 및 구체적 하위 논리에 미치는 효과를 분석해 본 결과 논리적 사고력 향상에 효과적임을 확인할 수 있다.

## ABSTRACT

What matters in the society of knowledge and information is not that they just know certain facts, but that when faced with new situations, they should be able to develop novel ideas, apply them and do the problems or the tasks confronting them. This cannot be achieved through learning of mere knowledge. Computer programming lessons have shown positive effects on general thinking ability, metacognitive aspects, and logical thinking. The ACM has suggested that 8th grade students at the first level (second year students of middle school) be educated in programming languages such as LOGO to raise their ability to think logically. Previous studies have confirmed educational programming languages such as LOGO and BASIC, which are currently used, are helpful in improving the ability to think logically and to solve problems. However these languages lack connectivity with later learning. Little research has been done on 'Dolittle', an educational programming language, newly developed, using object-oriented notions. Few studies have been made of the effects of 'Dolittle' on the ability to think logically. The following results were obtained. The research didn't lead to a statistically significant improvement of the students' cognitive development level. However, proportional logic and combinational logic, among the six subcategories of logic, improved through the introduction of 'Dolittle' programming lessons. This leads to the conclusion that in the processing of solving the problems given, the students learned by themselves and improved their ability to think logically.

## 키워드

dolittle, logical, thinking

---

\* 안동대학교 교육공학과 박사과정

접수일자 2009. 02. 09

\*\* 안동대학교 컴퓨터교육과 교수

심사완료일자 2009. 04. 24

I. 서 론

지식정보사회에서 살아갈 사람들에게 중요시되는 능력은 어떠한 사실을 아는 것이 아니라, 새로운 상황에 당면했을 때, 새로운 아이디어를 개발하고 이를 적용해 주어진 문제나 과제를 해결할 수 있어야 한다[1]. 이것은 알고 있는 것과 이를 이용하여 문제를 해결해야 될 것 사이의 간격을 줄이기 위해 가장 적절하고 합리적인 방법을 찾을 수 있어야 한다는 것이다. 이는 단순히 지식의 암기로는 해결할 수 없는 일이기 때문이다.

ACM(Association for Computing Machinery)에서는 K-12(유아부터 우리나라 초등 6년, 중등 3년, 고등 3년에 해당)를 위한 컴퓨터과학 교육과정[2]을 구성하여 보고하고 있는데 초·중등 각 학년을 4단계로 나누어 각 과정에 필요한 교육과정이 자세히 소개되고 있다.

컴퓨터 프로그래밍이 일반적인 사고력 신장과 메타인지적 측면의 효과, 이해도에 대한 모니터링, 문제 분석 기술 등의 향상뿐만 아니라 컴퓨터의 계산 원리 이해와 논리적 사고의 표현 도구라는 학습 효과를 가지고 있는데, 이러한 이유를 토대로 ACM이 제시한 레벨 1에 해당하는 8학년(중학교 2학년에 해당) 학생들에게 논리적 사고력을 위해 로고(LOGO)와 같은 프로그래밍언어를 통한 교육이 필요함을 알 수 있다[14]. 현재 사용되고 있는 로고(LOGO)와 베이직(BASIC) 등의 교육용 프로그래밍언어(EPL: Educational Programming Language)는 논리적 사고 및 문제해결력 기능을 향상시킨다는 것은 선행 연구결과를 통해 확인되고 있다[3,4]. 그러나 새롭게 개발된 교육용 프로그래밍 언어인 ‘두리틀’이 논리적 사고력에 미치는 효과에 관한 연구는 좀처럼 찾아보기 힘들다.

따라서, 본 연구에서는 보다 효과적인 논리적 사고력 형성을 위해 현재 중학교 2학년의 인지 발달 수준을 파악한 후 교육용 프로그래밍 언어인 ‘두리틀’을 소개하고 이를 실제 수업에 적용하여, 인지 발달 수준에 미치는 효과와 논리적 사고력의 향상 및 구체적 하위 논리에 미치는 효과를 분석해 보고자 하였다.

II. 객체지향형 EPL ‘두리틀’

2.1 ‘두리틀’ 언어 소개

객체지향 프로그래밍 언어(OOPL : object-Oriented Programming Language)는 1960년대 Simula 프로젝트로 시작되었는데 여기서 말하는 객체란 실세계 객체와 마찬가지로 성질을 가진 엔티티(entity)이며, 고유의 방법으로 사건(event)에 반응하는 능력을 가진다는 개념이다. 또 프로그램은 일련의 객체로 이루어지며, 서로 다른 객체에 반응함으로써 동적으로 반응할 수 있다는 사상이다[5]. 1980년대에 이르러 객체지향 프로그래밍 언어는 기존의 프로그래밍언어에서도 객체지향적 특성을 확장하여 제안하기도 하였다[5].

‘두리틀’은 2001년 11월에 일본 쓰쿠바 대학의 가네 무네가 설계 개발하여 11월에 발표된 일본판 정식버전(V1.00)에 이어 2003년 고려대학교 연구팀의 한글화 작업을 통한 한글판 정식버전(V.1.16)이 나왔다. 그리고 현재 2006년 11월에 최신 버전(V.1.27)이 나왔다. ‘두리틀’은 자바(Java)로 기술되어서 윈도우, 매킨토시, 리눅스 등 학교 교육에서 행해지는 자바가 동작하는 여러 가지 플랫폼에서 동작시킬 수 있다.

‘두리틀’ 한국어 버전 1.26 실행 화면은 <그림 1>, <그림 2>와 같다. 프로그램 상단에 텍스트 표현으로 실제 프로그래밍을 할 수 있는 편집창과 이를 실행하였을 때의 화면인 실행창을 나타내는 두개의 탭이 있다.

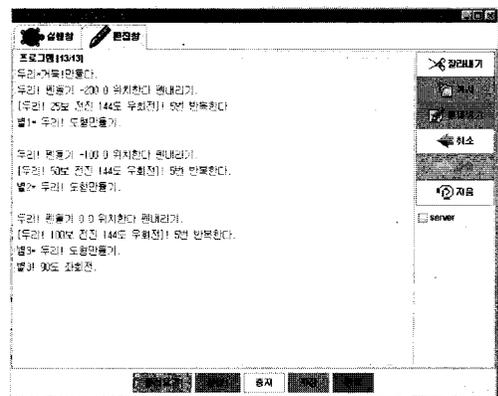


그림 1. ‘두리틀’ 편집창  
Fig. 1 ‘dolittle’ edit window

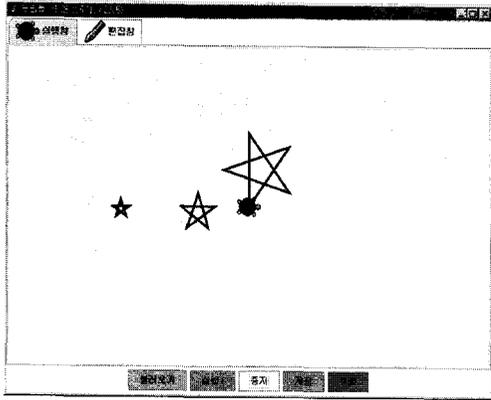


그림 2. '두리틀' 실행창  
Fig. 2 'dolittle' execution window

### 2.2 '두리틀' 특징

'두리틀'의 가장 큰 특징은 객체지향 개념 도입으로 상속이나 클래스와 같은 고도의 추상적인 개념 이해가 필요치 않는 범위 내에서 초중등 학생의 수준에 맞게 개발된 것이다. 일반 객체지향 프로그래밍 언어에서 사용되는 클래스 방식이 아닌 객체를 복사함으로써 원래 객체의 성질을 계승하는 프로토타입 방식으로 개념 이해가 쉬워서 교육용으로 적합하다.

또한 텍스트 표현에 기초한 소스 코드 작성으로 그래픽 프로그래밍처럼 시각적인 효과만의 강조를 지양하여 '계산기의 동작원리는 터득한다.'는 프로그래밍 교육의 원래 목표를 잘 반영하고 있다. 하나의 문장만으로도 동작을 관찰할 수 있는 간결함과 한국어 사용과 영어의 예약어가 일체 없음을 강조할 수 있다. 또한 수학 교육의 재귀개념을 위하여 복잡하고 이해하기 힘든 계층적 구문을 사용하는 것을 지양하여 2단계 이상의 구조를 가진 구문을 사용하는 대신 안쪽의 동작을 별도의 메소드로 분리하여 사용하게 된다[6].

### 2.3 '두리틀'과 로고의 비교

'두리틀'의 구문에서는 한국어의 변수명과 명령어를 갖추고 있어서 언어적인 장벽을 낮추고 프로그램에서 제공하는 기호를 적게 하는 것으로 습득을 용이하게 하고 있다. 이러한 구성에 의해 '두리틀'은 학습이 용이한 언어가 되었고 그 결과 학교 교육에서 취급하여 실제로 초등학교에서 고등학교의 초중등 교육, 그리고 대학에 있어서 고등교육에 있어서 수업에 이용되

고 있다.

수학, 과학 등 여러 분야에 널리 활용되고 있는 EPL인 로고와의 차이점을 살펴보자.

로고 언어가 구조적이고 절차적이어서 프로그램 구조가 복잡한 반면, '두리틀'은 객체지향형이기 때문에 라인이 짧아 간결하며 재사용성이 용이하다. 또한 한글 어순을 그대로 따르고 있으며 파생어가 아닌 언어 자체에 다국어 지원을 하고 있다. 반면 로고 언어는 파생 언어에서 한글을 지원하고 있으나 영문 어순을 따르고 있어서 EPL 한글화의 효과를 보지 못하고 있다. 또한 스프레드를 지원하지 않아서 타이머와 같은 고급 기능을 익히기에는 오랜 시간의 학습이 필요하다. 언어 자체에서 스프레드를 지원하는 '두리틀'을 이용할 경우 타이머 애니메이션을 아주 짧은 시간에 학습하여 활용 할 수 있다는 특징을 가지고 있다.

다음 <표 1>은 '두리틀'과 로고의 차이점을 정리한 것이다.

표 1. 로고와 '두리틀'의 비교  
Table. 1 Compare Logo and Dolittle

비교 \ 언어	로고	'두리틀'
형태	구조적, 절차적	객체지향
언어 지원	파생된 다른 종류의 언어	언어 자체에 다국어 지원
어순	영문 어순	한글 어순
스레드	지원 안함	지원 함
라인 길이	길다	짧다
프로그램 구조	복잡	간결(수정 용이)
재사용성	어려움 (상속 불가능)	용이(상속 가능)

## III. 인지발달과 논리적 사고

### 3.1 인지발달

피아제는 인지발달은 그 특성이 뚜렷이 나타나서 여러 단계를 거치며 그 특성은 중요 행동 양식을 나타내는 조작(operation)에 의하여 구별된다고 보고 있다. 조작은 출생 시부터 존재하는 것이 아니라 일생동안 4단계를 거쳐 점진적으로 구성되고 확장된다고 하였다. 발달 단계

는 실제로 독립적인 것이 아니며 앞 단계는 다음 단계를 위한 준비기간으로 볼 수 있고, 특정 단계를 나타내는 행동이 기대되는 연령의 범위도 고정적인 것이 아니다. 4 단계의 사고 중에서 본 연구와 관련된 구체적 조작기와 형식적 조작기의 특성에 대하여 살펴보면 다음과 같다 [7,8].

(1) 구체적 조작기 (7-11세)

이 시기에 이르면 아동은 논리적 조작(logical operation)이 발달한다. 즉, 그들은 구체적 문제에 적용될 수 있는 논리적 사고과정(조작)을 발달시킨다.

(2) 형식적 조작기 (11-15세)

피아제는 이시기를 인지 발달의 절정으로 보고 있다. 형식적 조작단계의 아동은 논리적 조작에 필요한 모든 종류의 문제를 해결할 수 있는 능력을 획득하게 된다. 그러나, 김동우의 연구[9]에서는 초등학교 6학년 280명을 대상으로 논리적 사고력 검사를 실시한 결과 구체적 조작기 수준의 아동이 62.9%, 과도기 수준의 아동이 30.7%, 형식적 조작기 수준의 아동이 6.4%로 조사되었다.

피아제가 제시한 인지 발달 단계에 의하면 초등학교 6학년 학생들은 형식적 조작기의 입문 단계에 해당되지만 선행 연구에서 조사된 집단은 거의 구체적 조작기와 과도기에 머무르고 있음을 알 수 있다.

3.2 논리적 사고의 개념

논리는 논리학에서 체계적으로 다루어진다. 논리학은 합리적 필연성에 의해 결론에 도달하는 타당한 추론의 학문이다. Piaget는 인간 지능의 근간을 사고로 보고, 그 사고는 문제해결 과정에 해답을 만들어내는 지적 조작으로 규정하였다[3].

사고라는 용어는 매우 넓게 쓰이는 용어이다. 가장 넓은 의미로는, 주어진 정보(자극)에 대해 무엇을 안다고 할 때 그 앎은 사고의 결과라고 말할 수 있다. 이처럼 넓게 정의하면 인지가 곧 사고이다. 즉 개념 형성과 언어 이해가 모두 사고라고 할 수 있다. 좁은 의미로 사고를 규정하면 그것은 추리, 판단 및 문제해결이다. 이러한 종류의 사고는 주어진 정보 이상의 것을 안다는 것을 뜻한다.

3.3 인지이론과 논리적 사고의 하위논리

Piaget 인지적 발달과정에서 학습자들은 연속적으로, 동일한 단계를 거치며, 새로운 사고의 수준을 통합하고 이전의 사고를 수정하는 방식으로 단계를 거치며, 새로운 사고의 수준을 통합하고 이전의 사고를 수정하는 방식으로 단계가 형성된다[1]. 또한 Piaget의 연구에 의하면 형식적 조작기에 도달한 학생은 보존논리, 비례논리, 변인통제논리, 확률논리, 조합논리의 사고가 가능하다고 한다.

(1) 보존논리(conservation logic)

어떤 수, 길이, 물질, 면적, 부피 등은 그 순서나 형태를 바꾸어 여러 가지 방식으로 제시한다 하더라도 항상 변하지 않는다는 것을 아는 능력을 의미.

(2) 비례논리(proportion logic)

어떤 두 비에 있어서 그 비의 값이 같다는 논리를 바탕으로 비례와 관련된 규칙 혹은 관계들 간의 관계(relations between relations)를 의미.

(3) 변인통제논리(control of variable logic)

상황의 모든 변인들을 인식하고 변인들의 역할에 관한 가설을 설정한 다음, 그 가설을 검증하기 위해 체계적으로 변인들을 통제하여 결론을 도출해 내는 능력을 의미.

(4) 확률논리(probability logic)

학습과정에 우연히 일어나는 사상 중에서 어떤 사상이 일어날 확률을 계산할 수 있는 능력을 의미.

(5) 상관논리(correlation logic)

두 개 변인 또는 현상사이에 어떤 상관적인 관계가 있을 것이라고 예상하는 능력을 의미.

(6) 조합논리(combination logic)

문제를 해결해 나가는 과정에서 있을 수 있는 여러 가지 경우를 빠짐없이 중복되지 않게 셀 수 있는 능력을 의미.

IV. 연구방법 및 절차

4.1 관련 연구 결과

전현식[10], 황건수[11], 이성근[12], 박원길[13]의 연구에서는 프로그래밍 교육을 통해 아동의 사고력을 신장시키기 위한 시스템을 개발하였다. 주로 비주얼 베이

직 프로그래밍 언어를 학습하면서 주어진 상황에 따라 문제를 설정하고 해결해 나가는 과정을 통해 문제해결력과 논리적 사고력을 향상시킬 수 있는 학습체제를 제공해 주고자 하였다.

현해경의 연구[3]에서는 LOGO 프로그래밍 수업이 논리적 사고에 미치는 영향을 알아보기 위해 중학교 2학년 학생을 대상으로 6차시 동안 수업한 결과 비례논리, 확률논리, 조합논리에 영향을 준 것으로 나타났다.

강혜진의 연구[1]에서는 초등학교 6학년들 대상으로 7차시 동안 프로그래밍 기초능력을 통해 아동의 논리적 사고력 신장에 관해 분석한 결과 변인통제를 제외한 보존논리, 비례논리, 확률논리, 상관논리, 조합논리의 향상을 가져왔다.

#### 4.2 대상 및 내용

ACM이 제시한 프로그래밍 교육시기와 선행연구결과를 토대로 하여 우리나라 중학교 2학년 학생 63명을 연구의 대상으로 6주간 6차시 수업을 실시하였다. '두리틀' 프로그래밍을 학습한 후 인지발달수준 및 논리적 사고에 미치는 효과와 논리적 사고의 6가지 하위논리에 미치는 영향에 대해 알아보고자 하였으므로, 실험 처치 전과 후에 인지발달수준과 논리적 사고력 검사를 실시하여서 어떠한 차이가 있는지 알아보았다. 또한 교수방법에 있어서 '두리틀' 프로그래밍 학습을 위해 채택한 과제는 본 연구자가 선정한 과제이며 수업내용을 세부적으로 살펴보면, 1차시의 수업내용은 '두리틀' 프로그램을 실행해 보고 기초 개념을 이해하고, 기본 명령어를 이해하는 수업이다. 2차시의 터틀 그래픽은 절차적인 프로그래밍의 기초개념으로 학생은 터틀 그래픽을 사용해서 화면에 삼각형 사각형의 도형을 그리는 것으로 프로그래밍을 체험한다. 입문단계에서 터틀 그래픽을 사용하는 이점은 프로그램이 정확하게 동작하고 있는가를 스스로 시각적으로 판단할 수 있기 때문에 올바른 동작에 이를 때 까지 스스로 실행결과를 피드백하여 디버깅 작업을 행할 수 있도록 하였다.

3,4,5,6차시는 객체 생성 및 조작으로 학생이 터틀 그래픽 객체에 분리된 도형을 만들기 위해 거북이를 이동하고, 그렸던 도형을 다시 객체로 만드는 과정을 통해 객체 생성을 학습하게 된다. 자세한 지도계획은 다음과 같다.

표 2. '두리틀' 프로그래밍 지도계획  
Table. 2 Dolittle programming lesson plan

차시	학 습 내 용
1차시	- 프로그래밍의 개념 파악 - 기본적인 '두리틀' 명령어
2차시	- 터틀 그래픽 도형 그리기
3차시	- 분리된 도형 그리기 - 함수의 블록화
4차시	- 객체의 개념 - 생성된 객체의 메소드를 통한 조작
5차시	- 함수의 응용 - 변수 사용
6차시	- 객체 탄생의 방법 - 객체 상속의 개념 - 객체 색칠하기

#### 4.3 검사 도구

본 연구에 사용된 논리적 사고력 검사 도구는 GALT (Group Assessment of Thinking)의 full Version과 How's your Logic Test 검사 도구를 사용하여 사전과 사후 다른 검사지로 검사하였다.

본 연구에 사용된 논리적 사고력 검사지는 6개 논리 영역 중 조합논리 영역만 4문항이고 나머지 영역은 2문항씩 14문항으로 구성되었는데, 1번에서 12번까지는 답과 이유를 묻는 이중선다형 문제이고, 11~14 문항은 문제에서 요구되는 모든 경우의 수를 쓰는 문제이다. 1번에서 12번까지는 답과 이유를 묻는 문제에 대한 답이 모두 맞을 때, 11에서 14번은 조합의 경우를 한 개까지 빠뜨린 경우만 정답으로 처리하였으며, 논리유형별 형성 정도와, 인지수준의 변화는 다음과 같이 판단하였다 [15].

##### 가. 논리 유형별 형성정도

GALT 개발자들이 제안한대로, 각 논리의 형성정도는 GALT를 이루고 있는 논리유형을 기준으로 각각을 형성(3점 부여), 과도기(2점 부여), 미형성(1점 부여)의 세 단계로 나누었다. 우선 각 논리에 해당하는 문항에 대한 답을 1·2·3·4로 구분한 후, 교차확인 방법에 따라 세 단계로 구분한 것이다. 다음 <표 3>은 1번 문항과 2번 문항을 예로 제시한 것이다.

표 3 논리 형성정도를 알기 위한 교차 확인  
Table. 3 Cross check for measure logical formative

문항	1				
	구분	4	3	2	1
	4				
	3				
	2				
1					

형성(3점) 과도기(2점) 미형성(1점)

나. 인지수준 변화 정도

채점된 전체 점수를 구하여 정답수가 5개 이하면 구체적 조작기, 6~9개는 과도기, 10이상이면 형식적 조작기로 판정하였다.

V. 연구결과 및 분석

5.1 인지수준의 변화

‘두리틀’ 프로그래밍을 학습한 집단은 학습하기 전보다 인지발달 수준에 유의미한 차이를 검정하기 위해 인지발달 수준에 대한 사전검사와 사후검사 결과 통계적으로 유의미한 변화는 없지만, 사전검사 결과 <그림 3>과 같이 구체적 조작기 25.4%(16명), 과도기 54.0%(34명), 형식적 조작기가 20.6%(13명)로 나타났다. 79.4%의 대부분의 학생이 과도기와 구체적 조작기에 머무르고 있음을 알 수 있다.

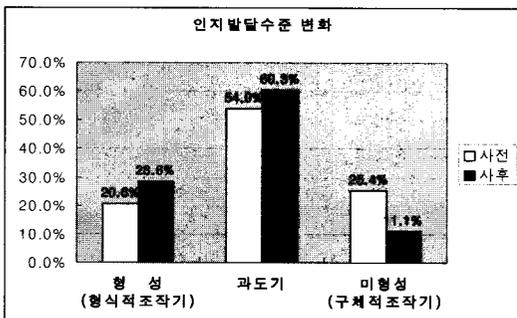


그림 3. 전체 학생의 인지발달 수준 변화  
Fig. 3 Cognitive development level change of Students

‘두리틀’ 프로그래밍 교육 후의 인지발달 수준 변화를 보면 구체적 조작기의 학생은 25.4%에서 11.1%(7명)로 감소하였고, 과도기의 학생은 54.0%에서 60.3%(38명)로, 형식적 조작기의 학생은 20.6%에서 28.6%(18명)로 증가했음을 알 수 있다. 따라서 구체적 조작기에 있는 학생에게 가장 큰 변화를 보였음을 알 수 있다.

5.2 논리적 사고의 형성정도

실험집단의 논리적 사고력 사전검사 평균이 1.95, 사후 검사 평균이 2.17로 나타났으며, 유의수준 (.05)에서 유의확률 .00으로 논리적 사고력에 있어 평균점수 사이에 유의미한 차이가 있는 것으로 밝혀졌다. <표 4> 결과에 의해 ‘두리틀’ 프로그래밍을 학습한 집단은 더 높은 논리적 사고력 신장을 보였다.

표 4. 실험집단의 논리적 사고 검사 대응 비교  
Table. 4 Comparison logical thinking of focus group

실험 집단	인원	평균	표준 편차	t-값	p-값
사전	63	1.95	.682	-3.869	.00
사후	63	2.17	.610		

5.3 하위 논리의 형성여부와 정도

<표 5>와 같이 ‘두리틀’ 프로그래밍을 학습한 후에는 논리적 사고의 하위 논리에 있어 비례논리, 조합논리에서만 사고력 신장이 있는 것으로 나타났다.

표 5. 논리적 사고의 하위논리 검사 대응 비교  
Table. 5 Comparison Sub-logical test

하위논리	실험 집단	인원	평균	표준 편차	t-값	p-값
보존논리	사전	63	7.71	.851	4.531	.000
	사후	63	6.84	1.696		
비례논리	사전	63	4.08	2.217	-5.600	.000
	사후	63	5.78	1.529		
변인통제 논리	사전	63	5.62	2.555	.151	.881
	사후	63	5.57	1.907		
확률논리	사전	63	6.16	1.961	-1.038	.303
	사후	63	6.38	1.782		
상관논리	사전	63	4.10	1.563	1.137	.260
	사후	63	3.81	1.401		
조합논리	사전	63	2.21	1.233	-5.799	.000
	사후	63	2.97	.803		

## VI. 결론 및 제언

현대는 정보화 사회에서 직면하게 되는 많은 문제들을 해결할 수 있는 능력을 요구한다. 본 연구는 이러한 시대적 요구에 따라 논리적 사고력의 중요성을 인식하고 이를 증진하기 위한 일부로 프로그래밍의 중요성을 인식하여 이에 적합한 교육용 프로그래밍언어인 '두리틀'(dolittle)을 소개 하고 있다. 또한 현재 중학생의 인지 발달 수준 정도를 파악하였다. 따라서 본 연구에서는 중학교 2학년 63명을 대상으로 6주 동안 '두리틀'을 활용한 프로그래밍 교육을 실시하고 중학생의 인지 발달 수준과 논리적 사고력에 미치는 영향을 분석하고자 하였으며, 지금까지의 연구 결과를 토대로 결론을 내려면 다음과 같다.

첫째, 인지 발달 수준에 있어서 과도기<형성<미형성 순으로 전체적으로 향상되었고, 구체적 조작기에 있는 학생 즉, 미형성 단계에 있는 학생에게 가장 큰 변화를 보였다. 따라서 '두리틀'을 활용한 프로그래밍 학습은 미형성 단계에 있는 초등학생들에게 더 많은 향상을 줄 것으로 기대된다.

둘째, '두리틀' 프로그래밍을 학습한 집단의 논리적 사고력의 형성 정도는 학습하기 전과 학습한 후에 유의미한 차이가 있었다( $t=-3.869, p<.05$ ). 즉, '두리틀' 프로그래밍을 학습하기 전보다 학습한 후에 높은 논리적 사고력 신장을 보였다.

셋째, '두리틀' 프로그래밍을 학습한 후 논리적 사고의 하위논리의 형성 정도가 다르게 나타났다. 6개의 하위 논리 유형 중 비례논리( $t=-5.600, p<.05$ )와 조합논리( $t=-5.799, p<.05$ )에서만 유의미한 차이가 있는 것으로 나타났다.

그러나 본 연구는 중학교 2학년에 한정하였고, 6주 동안 6차시 분만의 '두리틀' 프로그래밍 수업을 진행하였기 때문에 인지 발달과 논리적 사고력 향상에 충분한 영향을 주었으리라 단정할 수 없으므로, 본 연구 결과를 일반화하기에는 어려움이 있을 수 있다. 따라서 충분한 기간에 걸쳐 지속적으로 실험 처치에 관한 연구가 있어야 할 것이다. 또한 본 연구의 '두리틀'을 활용한 프로그래밍 교육이 형식적 조작기에 있는 중학교 2학년을 대상으로 이루어졌지만, 향후 연구에서는 형식적 조작기 입문 단계인 초등학생을 대상으로 연구가 되어야 할 것이다. 이러한 향후 연구를 통해, 프로그래밍 교육이 가장 적합

한 시기에, 아동의 현재 인지 발달 수준에 적합한 내용으로 교육이 이루어져야 할 것이다.

또한 '두리틀' 프로그래밍이 기존 교육용 프로그래밍 도구들과의 구체적인 비교 연구를 통하여 객체지향 '두리틀'의 차별성 연구 또한 이루어져야 할 것이다.

## 참고문헌

- [1] 강혜진, 프로그래밍 기초 능력 배양을 통한 아동의 논리적 사고력 신장에 관한 분석, 숙명여자대학교 교육대학원 석사논문, 2004
- [2] 길혜민, 객체지향형 교육용 프로그래밍 언어 'dolittle'의 적용과 평가, 컴퓨터교육학회 동계 학술대회, 2004
- [3] 현혜경, LOGO 프로그래밍 학습이 논리적 사고에 미치는 영향, 안동대학교 교육대학원 석사학위 논문, 2005
- [4] 류향미, LOGO 프로그래밍 학습이 논리적 사고 기능 향상에 미치는 효과, 연세대학교 교육대학원 석사학위논문, 1994
- [5] 하수철, 객체지향 프로그래밍 언어의 특성과 그의 비교, 정보처리학회 논문지 제2권 제2호, 1995
- [6] 김경미, 객체지향 언어 두리틀의 수학교육 활용, 고려대학교 수학교육학과 석사학위논문, 2004
- [7] 김경미, 완본과 축소본 GALT의 비교연구, 한국교원대학교 대학원 석사학위 논문, 1999
- [8] 김현재, 피아제의 이론과 임상법 실제, 배영사 신서, 1994
- [9] 김동우, 국민학생의 논리적 사고수준과 전류와 자기장의 학습 성취도, 한국교원대학교 대학원 석사학위 논문, 1992
- [10] 전현식, 박관우, 초등학생을 위한 비주얼베이식 교육시스템 개발, 한국정보교육학회, Vol.7 No.3, pp.299-308, 2003
- [11] 황건수, 설문규, 초등학교 고학년 아동의 사고력 신장을 위한 프로그램 언어 재량활동 교재의 개발과 적용, 한국정보교육학회 Vol.8 No.1 pp. 149-158, 2003
- [12] 이성근, 박관우, 문제해결력 향상을 위한 Visual BASIC 제어 문의 웹기반 학습모듈 설계, 한국정보

교육학회, Vol.8 No.1 pp.386-392, 2003

- [13] 박원길, 이재무, 아동과 초보자를 위한 프로그래밍 학습 시스템의 설계, 한국정보교육학회, Vol.5 No.2, pp.315-322, 2000
- [14] A model Curriculum for K-12 Computer Science :Final Report of the ACM K-12 Education Task Force Curriculum Committee 2003
- [15] Roadrangka, V., Yeany, R.H., & Padilla, M. J., The Construction and Validation of Group Assessment of Logical Thinking (GALT), Paper presented at the 56th annual meeting of the National Association for Research in Science teachong, Dallas, 1983

### 저자소개



권 참 미(Chang-mi Kwon)

2002년 동양대학교 컴퓨공학과  
학사  
2007년 안동대학교 교육대학원  
컴퓨터교육과 교육학석사

2008년~현재 안동대학교 교육공학과 박사과정



권보섭(Boseob Kwon)

1983 경북대학교 전자공학과 학사  
1990 충남대학교 전자공학과 석사  
1997 KAIST 전산학과 공학박사  
1983년 ~ 1985년: LG전자 연구원

1985~1998 ETRI 선임연구원

1998년 ~ 현재 안동대학교 컴퓨터교육과 교수

※ 관심분야: 컴퓨터교육, 차세대통신망, 센서네트워크