

# 중등 컴퓨터과학교육을 위한 객체지향형 EPL '두리틀'의 적용 및 평가

권대용<sup>†</sup> · 길혜민<sup>††</sup> · 염용철<sup>†††</sup> · 유승욱<sup>†</sup> · 兼宗 進<sup>†††</sup> · 久野 靖<sup>††††</sup> ·  
이원규<sup>†††††</sup>

## 요 약

제 7차 컴퓨터교과 교육과정은 소프트웨어 활용 부문에 치중되어 컴퓨터과학에 관한 원리교육이 이루어지지 못하고 있다. 한편, ACM K-12 컴퓨터과학 교육과정 모델에 관한 보고서에서는 현재의 컴퓨터교육이 그릇된 방향이며 컴퓨터과학 중심의 컴퓨터교육을 위해서 프로그래밍언어 교육을 통한 원리교육의 중요성을 강조하고 있다. 본 논문에서는 프로그래밍 체험학습을 위한 객체지향 기반의 새로운 교육용프로그래밍언어인 두리틀(Dolittle)을 소개한다. 두리틀은 간결한 구문과 한국어와의 대응성, 인크리멘탈 방식의 프로그래밍, 수치, 문자열, 배열과 같은 알고리즘과 구조화, 텍스트 위주의 프로그래밍 등의 설계방식을 적용한 언어이다. 중학생을 대상으로 실제 수업에 적용해 보고 학습효과를 분석한 결과, 객체지향기 반이지만 클래스나 상속 등의 개념을 모르고도 활용 가능한 교육용프로그래밍언어라는 것을 확인할 수 있었다. 또한 객체와 메소드 활용을 통하여 프로그래밍 결과를 쉽게 확인할 수 있고 오류 수정이 용이하므로, 학생들의 높은 흥미도를 유지하면서 학습성취도를 향상시키고 차후 학습으로의 연계가 수월하였다.

키워드 : 컴퓨터과학교육, 교육용프로그래밍언어, 객체지향형 EPL, 두리틀

## Application and Evaluation of Object-Oriented Educational Programming Language 'Dolittle' for Computer Science Education in Secondary Education.

Dae-Yong Kwon<sup>†</sup> · Hye-Min Gil<sup>††</sup> · Yong-Cheul Yeum<sup>†</sup> · Seoung-Wook Yoo<sup>†</sup>  
Susumu Kanemune<sup>†††</sup> · Yasushi Kuno<sup>††††</sup> · Won-Gyu Lee<sup>†††††</sup>

### Abstract

Current computer education is difficult to educate basic concepts and principals of the computer science because the 7th curriculum of computer education is focused on the application of software. According to the ACM K-12 report about the computer science education model, current computer education is taking the wrong way and we should put the highly priority on the education of the fundamentals through programming languages for a better computer education oriented to the computer science. This paper introduces a new object-oriented educational programming language "Dolittle". The design principals of Dolittle are simple syntax of Korean, incremental programming, text based programming, aliasing of function, and object-oriented programming. Being applied to middle school classes, we can confirm that Dolittle is easy to learn, and gives rise to high interest and keeps interest through a course, and also is of great practical use in class for programming novice.

**Keywords** : Computer Science Education, Educational Programming Language, Object-Oriented EPL, Dolittle

### 1. 서론

#### 1.1. 연구 배경 및 목적

† 고려대학교 대학원 컴퓨터교육학과  
†† 고려대학교 교육대학원 컴퓨터교육전공  
††† 一橋大學 総合情報處理 center  
†††† 筑波大學 大學院 Business 科學研究科  
††††† 고려대학교 사범대학 컴퓨터교육과(교신저자)

우리나라의 컴퓨터교육은 원리 이해가 목적인 컴퓨

터과학교육 보다는 소프트웨어 활용이 중심이 되는 컴퓨터소양교육으로 구성되어 있다[4]. 이는 전문적인 인력 양성의 부족과 급변하는 정보기술의 발전 동향에 발맞추기 힘든 결과를 초래할 수 있다[24].

ACM(Association for Computing Machinery)에서는 K-12를 위한 컴퓨터과학 교육과정[3]을 구성하여 보고하고 있는데 초·중등 각 학년을 4단계로 나누어 각 과정에 필요한 교육과정을 자세히 소개하고 있다. 이 보고서는 알고리즘, 컴퓨터 구조, 운영체제 등의 원리를 이해하기 위한 컴퓨터 과학교육 과목을 제시하고 있으며 특히 프로그래밍 언어 과목을 강조하고 있다.

또한 ACM이 제시한 교육과정에서는 레벨 1에 해당하는 8학년 (우리나라 중학교 2학년에 해당) 학생들에게 컴퓨터과학교육의 기본적인 목적인 알고리즘적 사고 형성과 문제해결력 습득을 위하여 LOGO와 같은 교육용프로그래밍언어(EPL: Educational Programming Language)를 사용할 것을 제안하고 있다[3].

현재 교육용 사용되고 있는 로고(LOGO)와 베이직(Basic)등의 프로그래밍언어는 이미 몇 세대 전의 언어로써 현대의 소프트웨어의 원리를 이해하는데 거리가 있으며 실제 교육의 적용 부분에 있어서도 학생의 흥미도와 학습 성취도가 떨어지고 차후 학습으로의 연계성이 상당히 부족하다.

따라서 이 논문에서는 보다 효과적인 컴퓨터과학교육을 위한 교육용프로그래밍언어 두리틀[12]을 소개하고 이를 실제 수업에 적용하여 교육 효과를 분석해 보고자 한다.

## 2. 컴퓨터과학교육

### 2.1. 컴퓨터과학교육의 필요성

컴퓨터교육은 크게 컴퓨터활용교육과 컴퓨터소양교육, 그리고 컴퓨터과학교육으로 분류된다[11].

현재 우리나라의 컴퓨터 보급률과 인터넷 보급률을 보았을 때 학생들은 아주 어렸을 때부터 필수적으로 컴퓨터를 접하게 되어 마치 아이가 사회로부터 자연스럽게 모국어 배우듯 컴퓨터를 사용하는 능력을 배우게 된다. 이러한 상황을 고려하여 보았을 때 단순히 컴퓨터를 사용할 수 있는 능력을 향상시키기 위한 컴퓨터활용교육과 컴퓨터소양교육은 현재의 학생들에게는 도움이 되지 못한다. 따라서 k-12 과정에서 컴퓨터교육으로써 컴퓨터활용교육과 컴퓨터소양교육은 적합하지 않다.

컴퓨터과학교육은 컴퓨터의 원리를 이해할 수 있는

컴퓨터과학에 대한 내용을 가르치는 교육이다. 이러한 컴퓨터과학교육을 통하여 학생들은 단순한 컴퓨터의 활용이 아니라 알고리즘적사고, 문제해결력 등 컴퓨터교육의 근본적인 목적들을 성취할 수 있다. 또한 ACM k-12 보고서에서도 “학생들은 컴퓨터과학을 다른 영역과는 별개로 일반적인 이해 수준을 증진시켜야 하며 21세기의 시민들은 적어도 그 사회에서 기능을 수행하기 위하여 컴퓨터과학의 원리를 이해한다[3].” 와 같이 컴퓨터과학교육의 필요성을 강조하고 있다.

### 2.2. 프로그래밍 교육의 중요성

컴퓨터과학교육에서 프로그래밍교육은 컴퓨터의 가장 중요한 원리인 알고리즘을 이해하는 도구로서 매우 유용하다.

컴퓨터는 알고리즘을 구현하는 기계이고, 정보는 알고리즘이 다루고 만들어 내는 대상이며, 프로그래밍은 알고리즘을 기술하는 수단이다. 학생들이 프로그래밍을 하는 가운데 보다 논리적이고 조직된 사고를 하게 되며 더 나아가서는 자기 자신이 어떻게 사고하는지를 탐색하게 된다. 이는 컴퓨터를 사고할 수 있는 대상(an object to think with)으로 보는 견해에 부합된다. 알고리즘 사고방식은 특별하고 강력한 사고방식으로서 복잡한 현대 사회에 대처하기 위해 필요하고 다른 과학 및 공학 분야에서도 매우 중요한 개념이다[2]. 따라서 학생들이 다양한 사고방식으로 새로운 형태의 문제에 접근할 수 있는 능력을 갖게 하는 것이 중요하며 이것은 프로그래밍을 통해서 효과적으로 교육할 수 있다.

## 3. 교육용프로그래밍언어

### 3.1. 교육용프로그래밍언어의 필요성

ACM에서는 “알고리즘적 사고를 위해서는 LOGO와 같은 교육용프로그래밍언어를 통한 프로그래밍 교육이 필요하다.”고 강조하며 K-8학년의 컴퓨터과학 기초를 위한 학습 단계에서부터 이러한 교육이 필요하다고 주장한다[3].

컴퓨터과학교육을 위해서는 프로그래밍 교육이 필수적임에도 학생들은 프로그래밍 학습을 매우 어렵다고 생각한다[1]. 프로그래밍은 문제해결력, 수학적 사고력, 효과적인 컴퓨터 사용능력, 프로그램 파일 생성, 컴파일, 결과 확인, 테스트, 버그 발견 및 수정 등의 복합적 기술을 모두 갖춰야 하는 작업이다. 따라서 학습자는

프로그래밍 과정에서 주어진 과제의 이해·분석, 과제 수행을 위한 방법 모색, 방법 실행 등의 학습 과정을 적용하기 힘들다. 인지적인 요인뿐만 아니라 프로그래밍 학습의 실패나 학습의 문제점, 학습에 대한 연구 등의 면에서 학생들이 프로그래밍 학습을 어려워하는 다양한 요인이 존재한다.

또한 프로그래밍 학습에 있어서 학습자는 알고리즘적 사고를 위한 도구로써 프로그래밍을 접하는 것인데 일반적인 프로그래밍 언어는 기본 문법과 구조를 이해하고 프로그램을 사용하는 방법을 익히는데 너무 많은 노력을 요구한다. 그래서 학습자가 학습 초기에 가지고 있던 의욕이나 동기는 상실되고 프로그래밍에 대한 부정적인 인식만 남게 되어 원리 교육을 받을 수 있는 차후 학습으로의 연계성이 결여된 채 프로그래밍 학습 자체를 포기하게 되는 것이다. 이러한 문제점을 해결하기 위하여 교육용프로그래밍언어가 필요하다.

### 3.2. 교육용프로그래밍언어에 요구되는 특징

교육용프로그래밍언어로서 요구되는 특징은 다음과 같다[12].

#### (1) 이해하기 쉽고, 습득 시간이 짧은 것 (이해가 쉬움)

개념과 구문이 간결하고 쉽게 배울 수 있을 것이 요구되며 클래스나 포인터 등의 어려운 개념을 학습해서는 안 된다.

#### (2) 시각적으로 확인해 가며 배울 것 (대화성)

입문용의 예제 프로그램은, 1행씩 늘어가면서 시험해 볼 수 있어야 하며 결과를 시각적으로 확인할 수 있어야 한다.

#### (3) 기본적인 프로그래밍 구조화 원리를 배울 수 있을 것 (구조화)

Bork[3]은 “구조화 프로그래밍을 행하는 것은 교육용 언어가 갖추어야 할 성질이다”라고 지적하고 있다. 기본적인 명령을 조합해서 프로그램을 작성하기 위해서는 구조화된 프로그램을 기술하는 것이 필요하다. 따라서 기본적인 알고리즘을 기술할 수 있어야 하며 프로그램을 구조화할 수 있어야 한다.

#### (4) 일상적으로 사용하는 소프트웨어의 원리와 결부될 것 (객체지향)

어플리케이션 소프트웨어는 GUI 화면에서 동작하는 것이 일반적이다. 일련의 동작을 하는 프로그래밍을

기술하기 위해서는 구조화에 덧붙여 객체 지향의 개념을 취급할 수 있는 것이 유효하다. 시각적인 GUI 객체를 쉽게 사용할 수 있어야 하며 GUI 객체를 조합해서 프로그래밍을 작성할 수 있어야 한다.

언어에서 제공되는 객체와 자신이 만든 객체를 부품처럼 조합하는 것으로써 높은 수준의 프로그램을 쉽게 기술할 수 있는 언어가 요구된다.

#### (5) 네트워크를 체험할 수 있을 것 (네트워크)

정보통신의 구조를 체험적으로 학습하기 위해서는 네트워크 통신을 쉽게 다룰 수 있는 언어가 필요하다. 따라서 네트워크 통신을 행할 수 있어야 하며 네트워크를 통하여 객체를 전송할 수 있어야 한다.

### 3.3. 교육용프로그래밍언어의 종류

일반적으로 잘 알려진 교육용프로그래밍언어로 로고(LOGO)를 들 수 있다.

로고 관련 보고서에서는 로고의 특징을 다음과 같이 정리하고 있다.

- 배우기 쉽다.
- 좋은 프로그래밍 습관을 증진시킨다.
- 프로시저 만들기과 재사용이 가능하다.
- 쓰기와 유지가 쉽다.
- LISP에서 파생된 인공지능 언어로 동적인 함수 범위를 지원한다.
- 재귀어이다.

말(MAL: Mathematical Algorithmic Language)은 수학교육을 위해 만들어진 로고에서 파생된 언어이다. 따라서 로고 언어와 마찬가지로 재귀적 알고리즘이 가능하여 수열의 점화식이나 프랙탈 그리기에 활용이 되고 있다. 현재는 MAL-LOGO, Java-MAL 등으로 확장되어 사용되고 있다[6].

스퀴(Squeak)은 Smalltalk-80으로부터 발전된 교육용프로그래밍언어이다[22][23].

스퀴는 완벽한 객체지향언어이며 강력한 멀티미디어 기능과 교육용으로 사용될 수 있는 Etoy를 제공한다. Etoy는 그림판과 같이 직접 이미지를 그린 후 그것을 객체화하여 조작할 수 있고 비주얼한 형태로 코딩을 할 수 있다. 유아부터 고급 프로그래머까지 폭넓게 사용할 수 있다. 현재 미국과 일본을 중심으로 과학과 수학, 컴퓨터과학교육에 사용되고 있다.

<표 1> 교육용프로그래밍언어(EPL) 비교

특징 \ 언어	로고 (LOGO)	말(MAL)	스킵
탄생시기	1969년	1989년	1995년
설계 목적 및 한계	조직화된 사고 체계와 문제해결능력 신장	베이직(Basic) 문법에 로고(LOGO)의 그래픽 기능 추가	컴퓨터활용, 소양, 과학교육을 포괄하는 도구
주요 활용분야	수학, 프로그래밍, 언어기술, 사회과학 등에 적용	수학적인 기호 체계와 문법으로 컴퓨터와 교육에 적용	수학, 과학, 컴퓨터과학 교육에 적용
교육적 가치평가	비교적 광범위하게 적용되고 있음	MAL-LOGO, Java-MAL 등으로 확장되어 수학교육에 적용	미국, 일본에서 활발히 적용되고 있음
학습 용이성	쉬움	쉬움	쉬움
객체지향	×	×	완벽한 객체지향 지원

위에서 살펴본 교육용 프로그래밍 언어는 각자 나름대로의 언어 설계 목적과 강점을 내세우고 있으나 문제 분석, 설계, 프로그램 작성과 실행, 오류 검토 및 수정의 4단계를 거쳐 얻을 수 있는 프로그래밍 언어 교육의 학습 효과, 즉 프로그래밍 과정에서 요구되는 문제 해결과 지속적 오류 검증 및 수정의 작업이 요구하는 반성적 사고를 통한 고등 인지 기술 향상[10]을 극대화 시키지 못하고 있다.

#### 4. 객체지향형 EPL ‘두리틀’

##### 4.1. 두리틀 언어 소개

두리틀은 텍스트 코딩을 기반으로 하는 객체지향형 교육용프로그래밍 언어이며 컴퓨터과학교육의 기본이 되는 프로그래밍을 위하여 개발 되어진 언어로써 [5][8][19] 성인교육[26], 수학교육[18], 로봇제어[9][21] 등 다양한 분야에서 활용중이다.

두리틀 설계 방침은 다음과 같이 다섯 가지이다. 첫째, 클래스를 사용하지 않고 오브젝트를 생성하는 프로토타입 방식의 객체지향을 채용한다. 이는 클래스의 정의에 대해 알 필요가 없고 클래스의 계층에 대해 이해할 필요 없이 복제를 통해 객체가 생성되므로 이해하기가 쉽다.

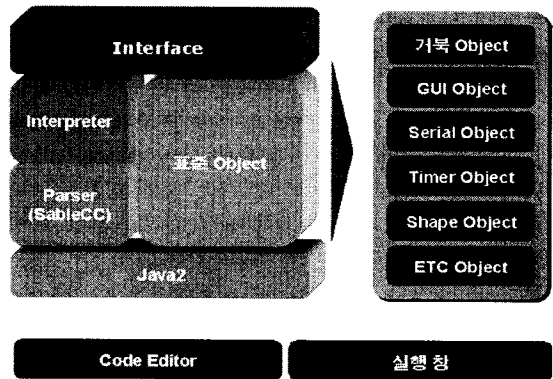
둘째, 메시지 송신에서 가스켓 송신을 이용한다. 이는 리시버를 반복해서 사용할 필요가 없기 때문에 프로그램이 간결해지고 같은 일을 하는 메시지를 한 눈에 알아 볼 수 있다.

셋째, 수식 등의 중치연산자를 블록 내에 단독으로

기술하도록 한다. 이것은 예약어가 존재하지 않기 때문에 사용자가 단어 선택에 제약을 받지 않는다. 또한 수식과 비교식 중에서 변수를 그대로 사용할 수 있다.

넷째, 메소드의 정의는 블록을 객체의 변수에 대입하는 것으로 한다. 이와 같이 함으로써 특별한 구문이 필요 없고, 가스켓 송신과 조합하면 대부분의 경우 1행으로 기술할 수 있다.

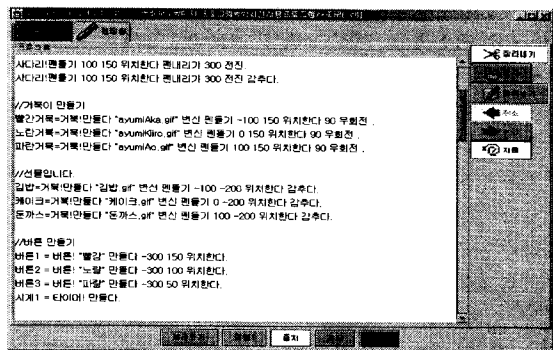
다섯째, 어린학생들도 쉽게 배우고 활용할 수 있도록 한국어 어순으로 되어있는 한글 프로그래밍을 지원한다[12][25].



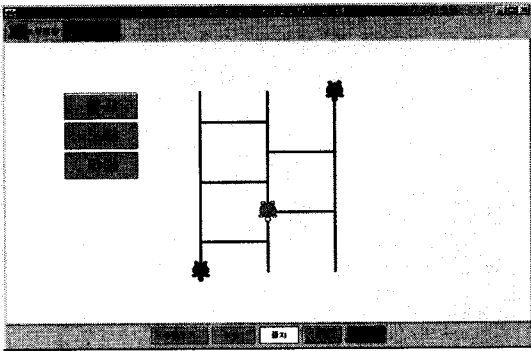
<그림 1> 두리틀의 내부구성

<그림 1>은 두리틀의 내부구성으로써 Java2 환경에서 동작하며, SableCC[15]에 의해 생성된 코드를 이용한 의미해석을 통한 구문해석, 인터프리터, 유저 인터페이스 및 표준 객체군을 갖추고 있다[12].

두리틀의 실행 화면은 <그림2>의 편집창과 <그림 3>의 실행창으로 구성되며, 우측에는 편집버튼과 하단에는 명령버튼으로 이루어져 있다.



<그림 2> 두리틀 편집창



<그림 3> 두리틀 실행창

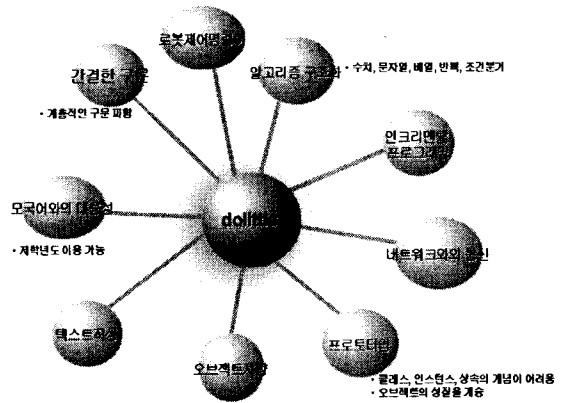
두리틀에서 사용된 구문은 <표2>와 같다.

<표 2> 두리틀 구문

프로그램 ::= (문'')...
문 ::= 대입문   메소드 정의   식
대입문 ::= 변수 '=' 식
메소드 정의 ::= 변수 '=' 블록
변수 ::= [항 ':'] 식별자
식 ::= 단순식   메시지 송신
메시지 송신 ::= [수신자] '!' 메시지
수신자 ::= 항
메시지 ::= 인수 메소드 이름 ([':'] 인수 메소드 이름)
인수 ::= 단순식
메소드 이름 ::= 식별자
괄호 ::= '(' 중치식 ')'   '(' 메시지 송신 ')'
단순식 ::= 수치 리터럴   문자열 리터럴   괄호   블록
블록 ::= ' ' '[' 식별자 ... ':' 문 (' 문' ...)'
중치식 ::= 중치식 연산자 중치식   항
항 ::= 단순식   식별자
연산자 ::= '+'   '-'   '*'   '/'   '>'   '<'   '>='   '<='   '=='   '!='
문자열 리터럴 ::= ""<인용부 이외의 임의의 문자> ... ""
' '<인용부 이외의 임의의 문자> ... ''
수치 리터럴 ::= [ '.' ]숫자 ... [ '.' ] 숫자 ...
식별자 ::= (영자) [한자](영자) [한]숫자 ...

4.2. 두리틀 특징

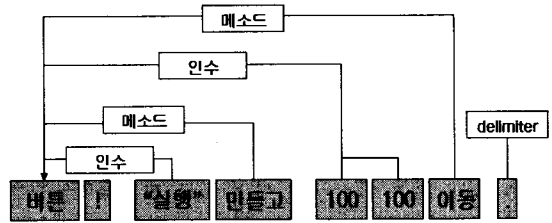
두리틀의 가장 큰 특징은 상속이나 클래스와 같은 고도의 추상적인 개념 이해가 필요치 않는 범위 내에서 객체지향 개념을 도입하고 있다. 객체를 복사하여 사용함으로써 원래 객체의 성질을 계승하는 방식을 이용하고 있다. 세부 적인 특징은 <그림 4>와 같다.



<그림 4> 객체지향형 교육용프로그래밍언어 두리틀의 특징

(1) 간결한 구문

계층적인 구문을 피하고 이해하기 쉬운 구문으로 되어 있다. 일반 프로그래밍언어와 달리 예약어 없어서 변수 사용에 어려움이 없으며 메소드 호출을 "!"를 사용하여 다음 <그림 5>와 같이 연속적으로 사용할 수 있다.



<그림 5> 구문 형태

(2) 한국어와의 대응성

영어의 예약어 등이 전혀 없고 기본적인 기호(괄호, 콤마, 마침표 등)를 포함한 한국어와 한국어의 어순으로 프로그래밍 된다. 또한 두리틀은 다국어어를 지원하며 소스의 한국어와 일본어는 프로그램에서 자동으로 변환이 가능하다[20].

(3) 인크리멘탈 프로그래밍

종래의 BASIC과 LOGO가 교육용언어로 사용되어져 왔던 요인의 하나는 인터프리터 방식으로 코딩 결과를 바로 확인할 수 있다는 점이였다. 두리틀도 이 이점을 계승해서 한 행만으로 그 나름대로의 동작을 기술할 수 있으며, 이것을 그대로 메소드로 정의할 수도 있다.

(4) 텍스트에 의한 기술

프로그래밍 및 문제해결 과정에서 이루어지는 고도의 논리적 표현을 위해서 두리틀은 텍스트 입력을 통해서 소스코드를 기술하고 있다. 더 나아가서 두리틀은 직접 코드 작성이 어려운 초등학교생들의 프로그래밍 교육을 위해서 비주얼 형태의 코딩을 할 수 있는 연구도 병행해서 진행되고 있다[16].

**(5) 알고리즘의 구조화**

수치, 문자열, 배열 등의 기본 데이터를 취급하고 반복과 조건분기 등의 제어구조를 갖추어 절차에 상당하는 메소드를 정의하는 것으로 구조화된 알고리즘을 기술할 수 있다. 이것을 통하여 계산기의 기초 개념을 체험하는 것이 가능하다.

**(6) 객체지향**

두리틀은 현대의 소프트웨어 개발 방법론의 핵심 주제인 객체지향 방식을 통해 프로그래밍을 할 수 있다. 학습자가 성취감을 느끼게 하기 위해서는 이와 같은 객체지향 개념이 적용된 교육용언어가 필요하다. 더불어 객체를 쉽게 파악하여 프로그램 구조 파악에 도움을 줄 수 있는 연구도 이루어지고 있다[17].

**(7) 프로토타입 방식**

직접 클래스를 생성할 수는 없으며, 제공되는 프로토타입의 객체를 사용하여 프로그래밍을 하게 된다. 이와 같은 프로토타입 객체를 이용한 학습은 학습자가 좀더 쉽게 문제 해결을 위해 객체를 사용할 수 있게 하여 교육적 효과를 높일 수 있다..

**(8) 네트워크와의 통신**

인터넷과 전자메일의 보급에 의해 컴퓨터의 활용이 네트워크와 떨어질 수 없다. 따라서 네트워크를 통한 데이터 교환 및 공유 프로그래밍을 경험 할 수 있다.

**4.3. 기존 프로그래밍언어와의 차이점**

두리틀은 오늘날 소프트웨어의 기술에 있어서 주요한 위치를 점하는 객체 지향의 사고방식에 기반한 교육용 프로그래밍언어이다. 기존의 BASIC과 LOGO 등이 플로우차트로 대표되는 「순서」로써 「계산기의 원리」를 기술하는 것과 비교하여, 두리틀은 동물과 버튼 등 실세계의 「사물」에 해당되는 「객체」로써 프로그램을 기술하고 있다. 이러한 방식은 인간과 가까운 사고방식으로 프로그램을 작성 할 수 있게 한다.

또한, 한국어의 변수명과 명령어를 사용함으로써 언어적인 장벽을 낮추고 프로그램에서 필요한 문법적 요

소들을 줄임으로 언어 습득을 용이하게 하고 있다. 그 결과 초중등 교육 및 대학과 같은 고등교육 수업에도 두리틀이 이용되고 있다[7][8][9][21].

다른 프로그래밍언어와의 비교는 <표 3>과 같다.

<표 3> 프로그래밍 언어와의 비교

부류	언어	이해하기 쉬움	대화성	구조화	객체 지향	네트워크
교육용 프로그래밍 언어	두리틀	O	O	O	O	O
	LOGO	O	O	O	X	X
	BASIC	O	O	X	X	X
범용 언어	Java	X	X	O	O	△
	C++	X	X	O	O	△
고등 교육용 언어	Blue	X	X	O	O	X
	若葉	O	O	O	X	X
비주얼 언어	MindStorm	O	O	X	X	X
	아르고블럭	O	O	X	X	X
예시 프로그래밍	KidSim	O	O	X	X	X
	Gamut	O	O	X	X	X

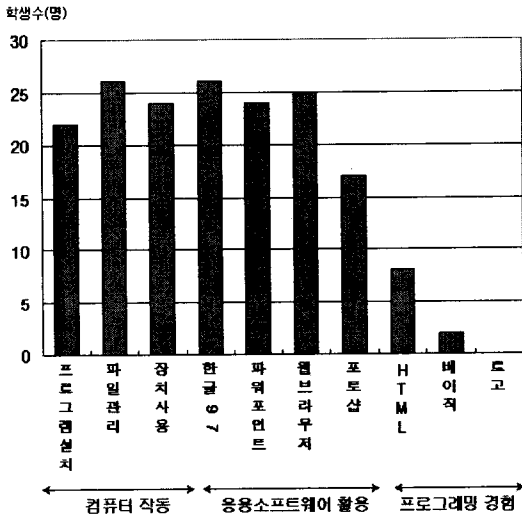
O : 항목을 만족한다.  
 X : 항목을 만족하지 않는다.  
 △ : 가능하지만 초보자에게는 쉽게 사용할 수 없다.

**5. ‘두리틀’의 중등 교육 적용과 평가**

**5.1. 교육 대상 분석 및 준비학습 내용**

중학교 1학년 여학생 26명을 대상으로 두리틀을 이용한 프로그래밍 교육을 실시하였다. 전원이 CA활동에서 컴퓨터부를 지원한 학생들이었고 전일제 CA시간을 포함하여 총 6차시에 걸쳐 수업이 진행되었다.

사전 설문 결과, 컴퓨터 활용 능력 면에서는 응용 소프트웨어 활용능력이 매우 우수하고 기본적인 운영체제 사용이 가능하였으나 문제점 발생 시 해결능력이 부족하여 부모님의 도움이나 A/S 기관을 이용한다는 의견이 90% 이상이었다. 프로그래밍 사용 경험은 전체의 약 7.7%인 2명이 베이직을 각각 1개월, 6개월에 걸쳐 사용해 보았으며 베이직 외의 프로그래밍 언어를 접해본 학생은 전혀 없었다. 결과는 <그림 6>과 같다.



<그림 6> 컴퓨터사용능력 및 선수학습에 관한 사전 설문 결과

프로그래밍 교육에 있어서 언어를 접하기에 앞서 프로그램을 다루는 방법 익히기에 너무 많은 노력을 요구하게 되어 학습 초기 의욕을 잃게 된다는 보고에 따라[1], 본 학습의 준비학습으로 실제 두리틀의 화면과 도구 버튼의 사용법, 그리고 간단한 실행 예를 설명함으로써 프로그래밍의 본래의 생소함(radical novelty)[1]이 최소화될 수 있도록 하였다. 또한 터틀 그래픽에 관한 텍스트를 배부하여 간단한 예를 접함으로써 외재적 동기 부여를 강화하였다.

5.2. 교육 내용 및 평가

총 6차시동안 각 차시마다 강의식의 학습 내용 전달과 학생 실습으로 이루어진 부분에 대한 시간 분배에 주의를 기울여 수업을 진행하였다. <표 4>는 전체 학습 지도안이다. <표 5>는 4차시에 대한 세부지도안이다.

<표 4> 두리틀 중등교육 적용 학습 지도안

학습주제	차시	주요 학습내용 및 활동	지도시 중점사항
터틀 그래픽	1/6	·두리틀 프로그램 실행 ·터틀 그래픽 도형 그리기	TP 자료를 이용하여 두리틀 프로그램 설치 및 사용 방법을 설명하고 간단한 예제 프로그램 작성 및 실행결과를 보여준다.
도형 객체 생성·조작	2/6	·객체의 개념 ·생성된 객체의 메소드를 통한 조작	메소드를 통해 생성한 객체의 개념을 이해시키고 객체의 조합으로 새로운 객체를 생성하는 과정을 꼼꼼하게 설명한다.
객체 활용 작품	3/6	·예제 프로그램 감상 및 소스 분석 ·객체 조작을 통한 작품 작성	디버깅하는 과정을 도와주고 작품 결과에 대한 피드백을 준다.
타이머 애니메이션	4/6	·타이머의 개념 ·도형 객체를 타이머 메소드를 이용하여 애니메이션 ·다양한 예제를 통한 실습	타이머 문법을 정확히 설명하여 잘 활용할 수 있도록 하고, 다양한 예제를 통해 창의적인 프로그램 작성이 될 수 있도록 지도한다.
GUI 부품 활용	5/6	·GUI 부품 체험 ·버튼을 이용하여 객체를 조작 ·다양한 예제를 통한 실습	버튼이나 텍스트 필드 등의 GUI 부품을 설명하고 이를 이용하여 그림판 등의 간단한 소프트웨어 제작을 할 수 있도록 지도한다.
자유 과제	6/6	·간단한 소프트웨어 제작 ·학생 작품 감상 및 평가	작품 발표 후 평가를 주고 평가서를 통해 학생 서로간의 평가가 이루어지도록 한다.

평가는 아래와 같은 항목으로 이루어졌다.

- (1) 사전, 사후 설문지9
- (2) 매 시간마다의 설문지
- (3) 2회 형성평가
- (4) 학생 작품

본 학습을 시작하기 전에 작성한 설문지와 모든 학습이 끝난 후에 작성한 설문지 내용을 비교하여 프로그래밍 학습에 대한 전반적인 이해도와 컴퓨터 원리 학습 정도 차이를 분석하였다. 또한 매 시간마다 설문지 작성을 통해 어려움과 즐거움, 과제 해결에 있어서의 학습성취도 정도를 분석해 봄으로써 변화 정도를 알아 보았다. 2번의 지필식 형성평가는 5문항 각각에 20점을 분배하여 100점 만점을 기준으로 평균점을 분석해 봄으로써 학습 내용에 대한 세부 항목 이해도를 평가 하였다. 마지막으로 학생 작품 분석을 통해 두리틀의

활용능력 및 개념 이해도를 평가하였다.

<표 5> 두리틀 세부지도안 (4차시 타이머 애니메이션)

학습자료	교사		학생		
	지도안, TP자료				
구분 단계	주요 학습내용	교수-학습 세부활동 내용		시간	지도상 유의사항
		교사 활동	학생 활동		
예비시간	·예비 시간을 갖는다.	·실습실 환경 정돈 ·인사	·지정된 좌석에 앉아 컴퓨터를 켜 다		
도입	·선수 학습 내용 ·학습 목표 제시	-지난 시간에 작성한 프로그램을 실행하여 명령어 사용법 등의 내용 을 다시 한번 숙지시킨다. -본시 학습목표를 제시한다.	-저장되어 있는 프로그램(지난 시 간 자신이 작성한 것)을 실행시켜 결과를 확인한다. -본시 학습목표를 숙지한다.	5	
전개 (1)	·타이머 예제프로그램 ·타이머란? ·타이머 문법 ·애니메이션	-타이머 예제 프로그램을 보여준 다. -타이머 개념을 설명한다. -타이머 문법을 설명하고 타이머 객체를 생성하게 한다. -애니메이션 프로그램을 작성하게 한다.	-타이머 개념을 이해한다. -문법을 익혀 간단한 예제 프로 그램을 작성한다. -실행결과를 확인하고 숫자를 달 리했을 때의 애니메이션 결과가 달라짐을 확인한다.	20	간단한 프로그 램 작성으로 문 법에 익숙해질 수 있도록 지도 한다.
전개 (2)	·타이머 고급 예제프로 그램	-타이머를 이용한 작품을 보여준 다. -소스를 보여주고 내용을 설명한 다. -각자 아이디어로 비슷한 프로그램 을 작성해 보도록 한다.	-예제프로그램을 그대로 작성하 고 실행결과를 확인한다. -고급 예제프로그램 소스를 이해 한다. -타이머를 이용한 자신의 프로그 램을 작성하고 디버깅 과정을 거 쳐 실행결과를 확인한다.	15	수치의 변화에 따른 차이점을 설명한다.
정리	·학습내용 정리 ·설문지 작성	-타이머 애니메이션에 대한 본 차 시 학습내용을 정리한다. -학생들의 프로그램을 보여주고 평가한다. -본 차시에 대한 설문지를 작성하 게 한다. -다음 차시 학습내용을 예고한다.	- 다른 학생들의 작품을 보고 평 가한다. - 본인 프로그램의 장단점을 분 석해 본다. -설문지를 작성한다. -마침 인사를 한다.	5	

5.3. 교육 결과 분석

사전 설문 결과에서 학생들은 이론학습을 통해 프로그래밍 개념을 배웠지만 전혀 이해하지 못하고 있었다. 그러나 본 학습 종료 후 실시한 설문지에서 <표 7>과 같이 프로그래밍 원리를 세부 항목으로 표시하여 학생들의 이해도를 평가해 보았다. 두리틀을 학습한 본인이 「이해하고 있다」라고 느끼고 있는지를 학생에게 「자신이 이해하고 있다고 생각하는가」라고 하는 주관적인 설문을 실시하였다. 중학교 1학년 학생임을 고려해 수업 중에 ‘객체’라는 단어를 명시적으로 사용하지 않았고 ‘타이머’라는 단어 대신에 두리틀 한글 명령어대로 ‘시계’라는 단어를 사용하였으므로 설문 항목에서도 동일하게 적용하였다. 회답은 긍정과 부정 2가지를 두

었으며 설문 항목의 내용을 이해하지 못하거나 긍정과 부정 중간일 경우 무응답으로 처리하였다.

- 수업 중에 설명한 내용을 바탕으로 한 설문에서 프로그래밍의 기초개념을 이해할 수 있었다라고 느끼고 있는 학생이 많다.
  - 수업에서 설명하지 않은 설문에 대해서는 약 60%의 학생이 「프로그래밍을 바르게 쓰면 자신이 생각한 대로 움직인다. 에러가 있으면 움직이지 않는다.」라고 경험적으로 인식하고 있다.
  - 타이머에 대한 설문에서는 수업 중에 설명하고 있지 않음에도 불구하고 「타이머가 비동기적으로 실행된다」라는 것을 알고 있는 학생이 약 75%이다.
- <표 7>은 학습 종료 후 설문결과이다.



<표 7> 본 학습 종료 후 설문 결과

설문 항목	긍정 (%)	부정 (%)	무응답 (%)
컴퓨터는 자신이 작성한 프로그램 대로 정확히 동작한다.	63	22	15
프로그래밍은 정해진 규칙이 있고, 그 규칙대로 동작한다.	58	30	12
거북 명령어를 통해 서로 다른 이름을 가진 여러 마리의 거북을 만들어낼 수 있다.	92	8	0
시계에 의한 동작은 정해진 규칙 외에 자신이 설정한 시간대로 실행된다.	75	12	13
진진과 후진의 거리와 도형의 색을 자신이 지정할 수 있다.	89	0	11
같은 구문을 여러 번 사용할 때는 괄호를 이용하여 반복 명령어를 이용한다.	53	8	39
자신이 만든 객체를 버튼을 이용해서 조작할 수 있다.	73	16	11

매 시간 수업 종료 시 수업의 어려움과 즐거움, 성취도를 설문지로 작성한 결과가 <그림 7>과 같다. 수업이 진행되면서 어려움이 증가하고 있으나 즐거움이나 성취도는 높은 수치를 유지하고 있고, 실제 수업 시간인 45분 동안 학생들의 집중도가 높은 편이었다.

- 수업이 진행됨에 따라 즐거움이 증가했다.

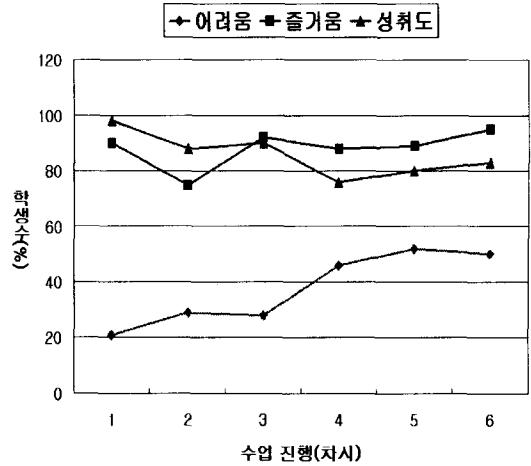
즐거움을 보면 「즐거웠다」라고 답한 학생의 비율은 수업이 진행되는 것에 따라서 전반부의 최저 75%에서 후반부 최저 88% 최고 95%로 증가하였다. 이것은 수업을 전반부와 후반부로 나누어서 행했던 결과로부터 의미 있는 변화이다.

- 달성했을 때의 즐거움이 증가했다.

어려움과 즐거움을 보면 수업진행에 따라서 어렵다고 느끼는 비율이 증가하고 있는데 즐거움을 느끼는 학생의 비율도 증가하고 있다. 과제의 달성도가 즐거움에 관련되어 있다는 것을 알 수 있다.

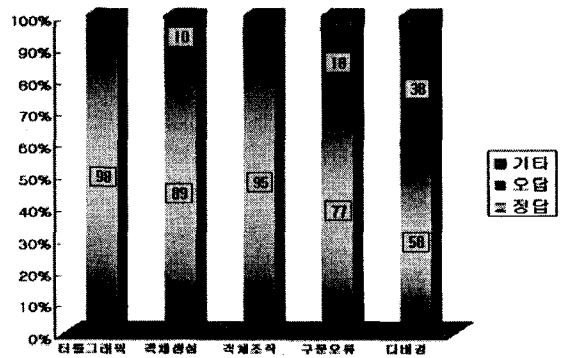
- 난이도가 높은 시간이 있었다.

어려움을 보면 타이머를 취급한 4차시의 난이도가 높게 되고 있다. 5차시에서 GUI 부품 활용과 응용에 있어서 어려움을 느끼는 학생이 증가하고 있고 타이머와 GUI 부품을 자신의 작품에 응용하는 것이 어려웠던 것을 알 수 있다.

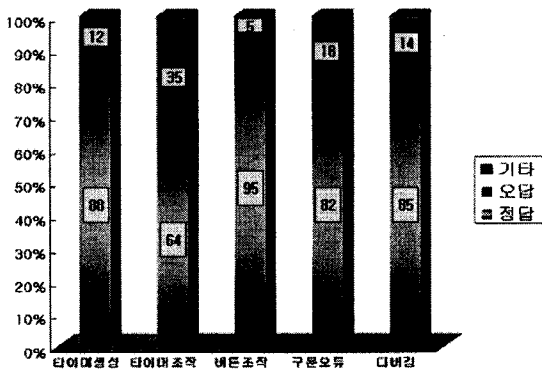


<그림 7> 매 차시마다 진행된 설문 결과

두 번에 걸쳐 시행된 형성평가는 각 5문항으로 구성하였으며 1, 2차시 수업과 4, 5차시 수업 내용에서 가장 기본적인 개념의 이해도를 평가할 수 있도록 하였다. 평가 결과, 학생들은 터틀 그래픽을 이용한 거북 객체 생성과 조작이 가능하며 기본적인 문법을 익히고 있었고, 반복의 정확한 문법은 작성하지 못했지만 그 개념은 이해하고 있었다. 또한 구문 에러를 발견하여 디버깅할 수 있고 객체에 메소드를 정의하여 활용할 수 있는 것으로 나타났다. 평가 결과는 <그림 8> <그림 9>와 같다.



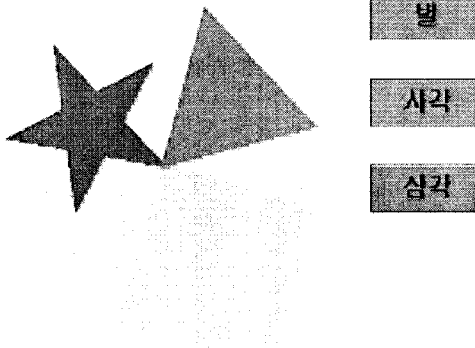
<그림 8> 1차 형성평가 결과



<그림 9> 2차 형성평가 결과

3차시에 터틀그래픽과 객체를 통한 간단한 프로그램을 작성하고 본 학습의 강의식 내용이 모두 종료하고 난 후 6차시에 학생 개인 단위로 간단한 프로그램을 작성하게 하였다.

학생들의 대부분은 전 차시에 보여준 예제 프로그램과 비슷한 프로그램을 작성하였는데 버튼 조작으로 선과 도형을 그리고 타이머 애니메이션을 이용한 작품을 완성하였다.



<그림 10> 학생작품 예

<그림 10>은 학생작품의 예를 표시한다. 이 프로그램에서는 화면상의 버튼이 배치되어 있고 각각의 버튼에 다음 종류의 메소드가 정의되어 있다.

- 터틀그래픽 이용(삼각, 사각, 별)
- 도형그리기 버튼  
(반복 명령으로 도형 객체를 생성, 색 지정 명령)
- 타이머 이용한 애니메이션  
(타이머의 비동기적 실행을 이해하고 있음)

전 차시 수업을 통해 배운 개념을 이해하여 활용할 수 있는 수준이었다.

## 6. 결론

프로그래밍 교육이 컴퓨터과학교육에서 매우 핵심적인 요소이고, 이를 학교 교육과정에 적용하기 위해서는 누구나 쉽게 배우고 익힐 수 있는 교육용 프로그래밍언어가 필요하다. 본 논문은 이에 적합한 객체지향형 교육용프로그래밍언어 두리틀(Dolittle)을 소개하고, 이를 학교 교육에 적용하여 그 결과를 분석하였다.

실제 우리나라 중학교 1학년 학생들을 대상으로 두리틀을 프로그래밍 교육에 적용해 본 결과 다음과 같은 몇 가지 특징을 찾아볼 수 있었다.

첫째, 객체지향 기반이면서도 상숙이나 클래스 등과 같은 고수준의 개념 이해를 필요로 하지 않아 초중등 교육에서도 적합하다.

둘째, 구조가 간단하고 객체와 메소드 활용을 통해 프로그래밍 결과를 쉽게 확인하고 오류 수정이 용이하므로 학생들의 흥미도와 관심이 아주 높다.

셋째, 버튼이나 애니메이션 효과 등으로 그림판과 같이 간단한 소프트웨어를 작성하기 쉬워서 학업성취도와 차후 학습으로의 연계성이 우수하다.

본 연구에서는 이러한 특징들을 갖고 있는 두리틀을 프로그래밍 교육에 적용한 결과, 누구나 쉽게 흥미가 유지되는 가운데 프로그래밍 학습이 이루어질 수 있음을 확인하였다.

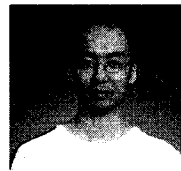
본 논문에서는 두리틀이 학교교육과정에서 입문자를 위한 프로그래밍 교육에 적합한 언어임을 확인하였으나, 이후 연구에서는 두리틀과 다른 교육용프로그래밍언어를 동일한 학습에 적용하고 그 결과를 비교 분석하는 방향으로 연구 범위를 확장하고자 한다.

## 참 고 문 헌

- [1] Tony Jenkins, "On the difficulty of Learning to Program", 3rd Annual LTSN-ICS Conference, pp.53-58, 2002.
- [2] Judith Gal-Ezer, David Harel. Erstpublikation, "What Else Should CS Educators Know?", CACM Vol.41, No.9, pp.77-84, 1998.
- [3] "A Model Curriculum for K-12 Computer Science", Final Report of the ACM K-12 Education Task Force Curriculum Committee, <http://www.acm.org/education/k12>, 2003.
- [4] 신은미, "고등학교 컴퓨터과학 교육을 위한 교육과정 연구", 고려대학교 석사학위 논문, 2002.
- [5] 兼宗進, 御手洗 理英, 中谷 多哉子, 福井 眞吾, 久野 靖. "學校教育用オブジェクト指向言語「ドリトル」

- の設計と實裝(학교교육용 객체지향언어 「두리틀」의 설계와 구현”. 情報處理學會論文誌, Vol.42, No. SIG11(PRO12), pp.78-90, 2001.
- [6] 서울대학교 사범대학 교육매체제작실, <http://edunet4u.snu.ac.kr/>
- [7] Susumu Kanemune, Shingo Fukui, Yasushi Kuno, Takako Nkatani, Rie Mitarai, "Dolittle-Experiences in Teaching Programming at K12 Schools", 2004.
- [8] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖. "初中等教育におけるオブジェクト指向プログラミングの實踐と評価(초중등교육에서 객체지향형 프로그래밍의 실천과 평가)", 情報處理學會論文誌, Vol.44, No.13, pp.58-71, 2003.
- [9] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖. "화면을 뛰쳐나온 객체(Object): 자립형로봇을 활용한 정보교육의 제안", 情報處理學會論文誌, Vol.42, No.6, pp.1610-1624, 2003.
- [10] 조미현, "프로그래밍의 학습 효과 및 교수 방법, 교육개발", 한국교육개발원, 1992.
- [11] 김현철, "정보 컴퓨터 소양교육의 현황과 개선 방향", 한국정보과학회지, 21(9), 2003.
- [12] 兼宗進, "教育利用を目的としたオブジェクト指向言語の研究", 츠쿠바대학 박사학위논문, 2003.
- [13] 오정석, 정종인, "제7차 컴퓨터 교육과정의 문제점과 해결책에 관한 연구", 컴퓨터교육학회논문지, Vol.7, No.3, 2004.
- [14] 표창우, "프로그래밍 언어 교육 현황 및 개선 방향", 정보과학회지, Vol.16, No.9, 1998.
- [15] SableCC, <http://sablecc.org>
- [16] 박소영, 권대용, 염용철, 유승욱, 이원규, "객체지향형 교육용프로그래밍언어 '두리틀' 시각화 프로그래밍 지원 시스템", 컴퓨터교육학회 하계 학술대회, pp.403-408, 2004.
- [17] 장혜선, 권대용, 염용철, 유승욱, 이원규, "객체지향형 교육용프로그래밍언어 '두리틀'을 위한 객체트리 설계 및 구현", 컴퓨터교육학회 하계 학술대회, pp.399-402, 2004.
- [18] 김경미, "객체지향형 교육용프로그래밍언어 '두리틀(Dolittle)'의 수학교육 활용", 고려대학교 석사학위논문, 2004.
- [19] 김혜민, 염용철, 유승욱, 이원규, "객체지향형 교육용 프로그래밍 언어 'dolittle'의 적용과 평가". 컴퓨터교육학회 동계 학술대회, 2004.
- [20] Susumu Kanemune, Wongyu Lee, Yasushi Kuno, "International Collaboration in School Based on Multilingual Programming Language", IPSJ 2004(9), pp. 7-12, 2004
- [21] Shuji Kurebayasi, Susumu Kanemune, "Comparison between robot control programming and application programming in a junior high school", IPSJ 2004(9), pp. 21-28, 2004
- [22] Squeak, <http://squeak.org/>
- [23] Smalltalk, <http://www.smalltalk.org>
- [24] 이원규, 정효숙 초중등학교에서의 컴퓨터과학교육의 역할과 필요성, 한국정보과학회지, Vol.22, No.5, pp.31-34, 2004
- [25] 최혜심, 염용철, 유승욱, 이원규, "객체지향형 교육용프로그래밍언어 두리틀의 다중 예약어 지원 체계", 컴퓨터교육학회 하계 학술대회, pp.365-370, 2004.
- [26] 최은조, 염용철, 유승욱, 이원규, "성인교육에 있어서 객체지향형 EPL 두리틀 적용 및 분석", 컴퓨터교육학회 하계 학술대회, pp.371-374, 2004.

## 권대용



2003 고려대학교 사범대학  
컴퓨터교육과(이학사)

2004~현재 고려대학교 대학원  
컴퓨터교육학과 석사과정

관심분야: 컴퓨터교육, EPL, 교육용 로봇  
email: dykwon@comedu.korea.ac.kr

## 김혜민



2002 성신여자대학교 전산학과  
(이학사)

2004 고려대학교 교육대학원  
컴퓨터교육 전공 (교육학석사)

관심분야: 컴퓨터교육, EPL  
email: aromagil@hanmail.net

### 염용철



1991 서울교육대학교 수학교육  
(교육학학사)  
2001 서울교육대학교  
컴퓨터교육과(교육학석사)  
1991~1995 서울개봉초등학교  
교사

1996~1999 서울개명초등학교 교사  
2000~2003 서울세곡초등학교 교사  
2004~현재 고려대학교 대학원 컴퓨터교육학과  
박사과정  
관심분야: 컴퓨터교육, EPL  
email: yycok@comedu.korea.ac.kr

### 久野 靖



1984 東京工業大學大學院 수료  
1986 東京工業大學 理學博士  
1989 筑波大學 社會工學系 講師  
1990 筑波大學 助教授  
2000~현재 筑波大學 大學院  
비즈니스科學研究科 교수

관심분야: 프로그래밍 언어, EPL  
email: kuno@gssm.otsuka.tsukuba.ac.jp

### 유승욱



1983 충남대학교  
기계교육학과(공학사)  
1985~1989 자양중학교 교사  
1989~1992 용곡중학교 교사  
1992~1996 종로산업학교 교사  
1996~현재 강남공업고등학교  
교사

2002 고려대학교 교육대학원 컴퓨터교육전공  
(교육학석사)  
2002~현재 고려대학교 대학원 컴퓨터교육학과  
박사과정  
관심분야: 컴퓨터교육, EPL  
email: swyoo@korea.ac.kr

### 이원규



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

1993~1995 한국문화예술진흥원 책임연구원  
1996~현재 고려대학교 사범대학 컴퓨터교육과  
교수  
관심분야: 컴퓨터교육, 정보검색, 데이터베이스  
email: lee@comedu.korea.ac.kr

### 兼宗 進



1987 千葉大學 電子工學科  
1989 筑波大學 大學院  
理工學研究科 (공학석사)  
1989 (주)Ricoh 소프트웨어연구소  
2004 筑波大學 大學院  
비즈니스科學研究科  
2004~현재 一橋大學  
總合情報處理센터 助教授

관심분야: 프로그래밍 언어, EPL  
email: kanemune@cc.hit-u.ac.jp