

비주얼 베이직을 이용한 초등 정보과학영재용 프로그래밍 언어 교육내용 탐색

강성원(인천산곡초등학교)

ksungwon@dreamwiz.com

이재호(경인교육대학교)

jhlee@ginue.ac.kr

요 약

초등 정보과학 영재교육의 핵심이 되는 프로그래밍교육은 창의적 사고와 수준 높은 문제해결력을 동원하여 새로운 알고리즘을 생성해내는 정보 교육의 중요한 분야임에도 불구하고 초등 정보과학영재의 특성을 고려한 체계적인 교육 내용이 확립되어 있지 못한 상태이다.

본 연구에서는 초등 정보과학영재를 위한 프로그래밍 교육의 필요성을 탐색하여 프로그래밍 교육내용 요소를 추출하였으며, 프로그래밍 언어 선정기준을 통해 비주얼 베이직을 교육용 프로그래밍 언어로 선택한 후 프로그래밍 교육내용을 제안하였다.

1. 서론

영재교육은 개인의 자아실현과 국가·사회적으로 인재를 양성하여 발전의 토대가 되게 한다는 면에서 그 중요성이 점점 높아지고 있다.

영재교육을 실시하는 목적은 국가발전, 그리고 학생들의 능력과 잠재력을 계발하기 위한 것 두 가지로 설명할 수 있다. 흔히들 첫 번째 목적인 ‘국가발전’ 측면이 강조되고 있는데, 이는 소수정예 고급인력을 양성해야 할 필요성에 대한 인식에서 강조되고

있다. 이와 함께 ‘개인의 역량에 맞는 교육기회 제공’이라는 측면에서 영재교육이 실시되고 있다는 측면을 간과해서는 안 된다. 이 경우 형평성을 골간으로 하는 공교육들을 유지하면서 그 보완으로 잠재력이 뛰어난 학생들을 대상으로 수월성을 가미하는 교육을 실시해야 한다는 인식에서 시작하고 있다(오승현, 2002).

현재 우리나라의 영재교육은 영재학교, 영재학급, 영재교육원의 영재교육기관에서 실시되고 있으며 수학, 물리학 등과 같은 순수과학 분야와 더불어 정보과학분야와 같은 다양한 분야에서 교육이 이루어지고 있다. 이 중 최근 각광받고 있는 정보과학분야의 영재교육은 순수과학 분야의 영재교육에 비하여 상대적으로 역사가 짧아 체계적인 연구가 이루어지지 않은 분야가 많고, 특히 초등 정보과학영재에 대한 연구는 미미한 실정이다.

나동섭은 영재에 대한 고찰을 통해 정보과학영재에 대한 정의를 제시하고 초등 정보과학영재의 교육을 위한 영역으로 기초기술소양교육, 지식윤리교육, 사고력신장교육, 프로그래밍교육의 4가지 영역을 설정하였으며, 각 영역에 대하여 세부 영역과 그에 따른 교육내용을 개발하고 이를 위한 교수-학습 전략을 제시하였다(나동섭, 2003).

이 중 프로그래밍교육은 창의적 사고와 수준 높은 문제해결력을 동원하여 새로운 알고리즘을 생성해내는 정보 교육의 중요한 분야임에도 불구하고 영재아를 위한 체계적인 교육 내용이 확립되어 있지 못한 상태이다(이재호, 2001).

이에 본 연구자는 초등정보과학영재를 위한 프로그래밍 언어 교육의 필요성을 알아보고 현재 사용되고 있는 프로그래밍 언어 교육 요소를 분석하고 공통내용을 추출하여 체계적인 교육 내용을 제시하고자 한다.

2. 초등 정보과학영재를 위한 프로그래밍 언어 교육

2.1 프로그래밍 언어 교육의 필요성

정보영재라 함은 수학적 문제해결 능력을 기본적으로 갖춘 영재임을 말하며, 여기에 창의적 문제해결 능력을 갖춘 영재를 정보영재라고 할 수 있다. 창의적 문제해결 능력이란 일반적인 지식과 지능기반을 토대로 확산적 사고와 행동, 논리적이면서도 비판적인 사고가 역동적으로 상호작용하여 새로운 산출물이나 해결책을 만들어내는

능력을 말한다. 특히 정보과학 분야는 창의적 문제해결 능력이 매우 중요시되는 분야이다. 창의적 문제해결 알고리즘을 세우고 컴퓨터 프로그래밍을 통해 그 결과를 확인해보게 되는 것이다(유철중, 2002).

컴퓨터 프로그래밍은 ‘컴퓨터를 활용하여 학습자가 컴퓨터에게 자신이 원하는 것을 수행하도록 하는 작업’이라고 정의되며, 컴퓨터 프로그래밍 교수/학습의 목적은 프로그래밍 언어의 습득 및 고등인지 기능의 습득이다.

프로그래밍 언어의 습득에 목적을 두는 것은 학습을 좁은 시각에서 본 것으로 프로그래밍 학습에 있어서 여러 명령어의 혼합된 사용에 관계된 규칙의 이해 등을 강조하는 것이므로, 특정 프로그래밍 언어의 명령어, 형식적 절차 및 변수 기능과 같은 일반적인 개념의 이해와 간단한 사용에 한정된다. 반면 고등인지 기능의 습득에 목적을 둔 것은 프로그래밍 과정에서 요구되는 문제 해결과 지속적인 오류 검증 및 수정 작업에서 요구되는 반성적 사고를 통한 고등인지 기술 향상을 강조한 것으로, 프로그래밍의 과정은 문제 해결 능력을 요구하므로 이 학습은 인간의 사고력을 향상시킬 수 있는 잠재적 특성을 지니고 있다는 전제이다.

컴퓨터 프로그래밍은 주어진 정보를 정확하게 체계적인 방법으로 구상하고, 그 정보에 대한 논리적인 조작을 할 수 있는 환경을 마련해 줌으로써 개인의 지적 능력의 계발을 도울 수 있다. 따라서, 컴퓨터 프로그래밍의 과정에서 학습자는 지식의 수용자라기 보다는 지식의 체제를 설계, 구축하는 입장에 놓이게 된다.

프로그램을 작성하는 과정을 대략적으로 보면 문제 분석 단계, 설계 단계, 코딩 단계, 시험 단계 및 유지 보수 단계로 나눌 수 있다.

Pea와 Kurland(1987)는 컴퓨터 프로그래밍에서 자주 사용되는 인지적 측면의 요구 사항을 다음과 같이 제시하고 있다.

(1) 처리 능력(Processing capacity) : 프로그래밍은 종종 많은 매개 변수들과 이에 배분되는 값들을 동시에 다루어야 하기 때문에 이러한 작업에 관련된 집중적 기억 능력과 정보 처리 능력이 요구된다.

(2) 유추적 추론 기능(Analogical reasoning skills): 유추적 추론 능력이 부족하면 프로그래밍과 관련된 지식과 능력을 지니고 있으면서도 이를 프로그래밍 작업과 관련시키거나 프로그래밍을 통하여 얻은 지식이나 기능을 다른 분야나 상황에 전이시키지 못한다.

(3) 조건적 추론 기능(Conditional reasoning skills) : 프로그래밍에서는 명령어의 반복적 실행, 입력 정보 점검과 검토와 같은 기능을 실행할 때 조건문을 사용하게 된다.

따라서 조건문을 이해하고 사용하는 것은 프로그래밍의 주요 부분이다.

- (4) 일시적 추론 기능
- (5) 절차적 사고(Procedural thinking)
- (6) 수학적 능력

컴퓨터 프로그래밍은 주어진 과제의 수행과 그 과정에서 당면하는 문제점의 해결을 위하여 소단위로 나누어진 단계적 작업들을 점차적으로 구축해 가는 작업이라 볼 수 있다.

이러한 프로그래밍이 갖는 독특한 특성 때문에 프로그래밍 학습의 학습 효과는 일반적인 사고력 신장, 메타 인지적 전략 획득, 이해도에 대한 모니터링 신장, 문제 분석 기능 습득 등에 효과가 있다고 할 수 있다.

특히 무엇보다도 초등 정보과학영재를 위한 프로그래밍 교육의 장점은 학생들에게 폭넓은 컴퓨터 소양을 심어준다는 것이다. 학생들은 프로그래밍 경험을 통하여 컴퓨터의 처리 과정에 대하여 이해하게 되고 컴퓨터 하드웨어에 대한 추상적인 개념을 확립할 수 있게 된다. 따라서 학생들은 어떤 상업적 소프트웨어가 잘 만들어진 것인지, 어떻게 이러한 소프트웨어들을 활용하여야 하는지, 자신이 원하는 문제를 컴퓨터를 활용하여 어떻게 해결할 것인지에 대한 식견을 가지게 된다.

2.2. 초등 정보과학영재 프로그래밍 교육의 문제점

초등 정보과학영재 프로그래밍 교육의 문제점을 알아보기 위해 '정보영재를 위한 교육내용에 대한 응답결과'를 살펴 보겠다(정두업, 2002).

<표 1> 정보영재를 위한 교육내용에 대한 응답결과 (다중응답)

구분 교육내용	컴퓨터 관련학과 교수전문가 24명(%)	컴퓨터/수학/과학 지도교사 47명(%)	정보영재교육 선발학생 105명(%)
프로그래밍 언어	11(46)	47(100)	94(90)
응용 프로그램 사용	3(13)	4(9)	6(6)
웹 설계 및 디자인	2(8)	15(32)	74(70)
창의력 개발 (문제해결능력)	23(96)	23(49)	69(66)
기타	1(4)	0(0)	0(0)

표에서 보면 교수·전문가 집단의 96%가 창의력 개발이 필요하다고 답변하였고, 46%는 프로그래밍 언어 교육이 필요하다고 답변하였다. 반면 컴퓨터/수학/과학 교사 집단은 47명 전원이 프로그래밍 언어 교육이 필요하다고 답변하였으며, 교수·전문가 집단과는 다르게 49%가 창의력 개발 교육이 필요하다고 답변하였다. 정보영재로 선발되어 교육을 받은 학생 집단의 90%가 프로그래밍 언어 교육이 필요함을, 66%가 창의력 개발 교육이 필요하다고 답변하였다.

이렇게 교수·전문가 집단과 교사 및 학생 집단의 응답 결과가 약간 다르게 나타난 이유를 유철중은 다음과 같이 해석하였다.

첫째, 교수·전문가 집단은 선발된 영재들이기 때문에 창의력(문제해결능력) 개발을 위한 교육을 주로 하면 큰 문제가 되지 않는 것으로 판단하는 것으로 보인다. 즉, 컴퓨터 프로그래밍 능력이 어느 정도 있는 것으로 생각하기 쉽고, 조금만 교육을 하면 쉽게 습득할 것이라는 생각을 하기 때문이다. 그러나, 학생들을 접해보면 이러한 생각은 현실과 너무 차이가 난다는 것을 알 수 있다. 말하자면, 영재성은 지녔더라도 초등학생이면서 프로그래밍 언어를 접하고 있기 때문이다.

둘째, 교사 및 학생 집단이 프로그래밍 언어의 교육이 필요하다는 것에 매우 높은 응답비율을 보이는 이유는 아무리 창의력 개발을 한다고 하더라도 결국 최종적인 해결 결과를 확인하기 위해서는 알고리즘을 작성하여 프로그램을 작성함으로써 가능하다고 생각하기 때문이다.

교사 및 학생 집단이 이렇게 생각하는 데는 큰 이유가 있다. 컴퓨터 프로그래밍은 수학, 물리, 화학, 생물, 지구과학 등과 같이 초등학교에서부터 그 기초를 이루는 공부를 꾸준히 해온 분야도 아니고 정규시간에 학습할 기회도 거의 없는 분야이기 때문이다. 따라서, 정보영재 교육에 참여하는 학생들은 극소수 학생을 제외하고는 다른 분야의 영재들과는 전혀 다른 학습환경에 처해있는 것이다. 우수한 프로그래밍 능력은 그 만두고라도 심지어 특정 언어의 컴파일 방법 등 기초적인 프로그래밍 학습 환경에 생소한 학생들이 대부분이다. 교사 및 학생 집단은 현장에서 이러한 현실을 절실하게 인지하고 있는 것이다.

컴퓨터를 이용하여 창의적으로 문제를 해결하기 위해서 우선적으로 해결해야 할 것은 기초 프로그래밍 능력의 함양이 절대적으로 필요하다는 것이다(유철중, 2002).

3. 프로그래밍 교육 내용 추출

프로그래밍 언어는 종류가 매우 많고, 일반적으로 잘 알려진 언어만 해도 수십 가지가 넘는다.

다양한 프로그래밍 언어들이 있지만 이 중에서 C언어, 비주얼 베이직, 자바스크립트, 액션스크립트(ActionScript)의 입문 서적과 프로그래밍 언어론을 분석하여 공통적인 교육 내용 요소를 추출하였다(오상진, 2003).

<표 2 > 프로그래밍 학습요소

구문	◆ 프로그래밍 언어를 지배하는 공통적인 규칙들
연산자 (operator)	◆ 연산자 - 단항연산자, 산술연산자, 쉬프트 연산자, 관계연산자, 비트 논리 연산자, 일반 논리 연산자, 3항 연산자, 대입연산자 등 ◆ 연산자 우선 순위
변수	◆ 변수의 정의, 선언 및 사용 - 변수 이름짓기(naming), 선언(declaration), 속성(attribute)의 지정(identify), 참조(reference), 값(value)
상수	◆ 상수 - 키코드, 수학 상수
수식	◆ 산술식, 논리식, 문자열식
제어문	◆ 조건문 - if문, else문, else if문, switch case ◆ 반복문 - while 문, for 문, do ~ while 문
자료형	◆ 형선언 ◆ 열거 자료형 ◆ 기본 자료형-수(number), 논리형(Boolean), 문자(Character) ◆ 구조적 자료형- 배열 ◆ 형 변환
함수	◆ 함수의 정의 ◆ 함수의 호출 ◆ 사용자 정의 함수

4. 프로그래밍 교육 언어 선정

4.1 프로그래밍 언어 선정 기준

초등정보과학영재를 위한 교육용 프로그래밍언어 선정기준은 다음과 같다.

첫째, 학습하기가 쉬어야 한다. 언어적인 속성을 기본적으로 가지고 있는 언어이면 서 다른 언어에 비해 학습하기가 수월하다면 그 언어는 초등정보과학영재에게 적합한 교육용 언어라고 볼 수가 있다.

둘째, 비주얼한 환경을 제공해야 한다. 비주얼한 환경이라는 것은 마우스를 이용해서 화면을 설계할 수 있으며 아이콘을 클릭 하는 과정과 풀다운 메뉴 등의 클릭을 통해서 코딩할 수 있는 환경을 제공해야 한다는 것이다.

셋째, 학습자에게 흥미와 동기부여가 잘 이루어질 수 있어야 한다. 같은 언어라고 할 지라도 학습 결과물이 보다 흥미롭고 동기 부여될 수 있는 여지와 가능성이 높다면 그 실습언어는 다른 언어보다 정보과학영재에게 적합하다고 볼 수 있다.

넷째, 학습자 수준과 관심에 부합되는 언어이어야 한다. 예를 들어 프로그래밍에 대한 기본 개념을 학습하는데 있어 부족하지 않은 두 가지 프로그래밍 언어가 있다고 할 때 그 중 학습자가 만들고 싶고 흥미와 관심을 가지게 할 수 있는 언어라면 비교 대상이 된 다른 언어에 비해 초등정보과학영재에게 적합한 언어라고 볼 수 있다.

다섯째, 도움말 기능이 잘 제공되어야 한다. 도움말 제공 기능은 코드 힌트, 코드 샘플 등이 저작환경에서 제공되어야 한다는 것이다. 물론 이러한 도움말은 반드시 한글로 제공되어야 한다(오상진, 2003).

4.2 초등 정보과학영재용 프로그래밍 언어로서의 비주얼 베이직

프로그래밍 언어 선정기준에 따라 본 연구자는 비주얼 베이직을 선정하였다.

물론, 이는 본 연구자 개인의 생각이며 다른 사람의 생각과 배치될 수 있다.

비주얼 베이직을 선정하게 된 이유로

첫째, 편한 인터페이스와 초보자가 배우기 쉽다는 것이다. 초등 정보과학영재도 초등학생이고, 프로그래밍에 생소함을 감안해야 할 것이다.

둘째, 빠른 학습 결과를 얻을 수 있다. 초등학생의 특성상 장시간의 프로그래밍 학

습은 무리가 있으므로 타 언어에 비해 개발 속도가 빠른 비주얼 베이직은 학생들의 학습 효과를 높일 수 있을 것이다.

셋째, 다양한 응용이 가능한 범용틀이다. 비주얼 베이직은 기초만 배우는 교육용 언어로서의 기능만 하는 것이 아니고 실제 다양한 분야에서 프로그램 개발도구로 사용되고 있다. 자신의 관심 분야(게임, 웹, 데이터베이스, 업무용 프로그램 개발 등)가 정해져 있다면 언어를 선택하는 기준이 좀 더 명확해 질 것이다. 하지만 자신의 진로가 정해져 있지 않은 초등학생에게는 범용틀로서의 비주얼 베이직이 의미를 가질 수 있을 것이다.

언어는 어떠한 기능을 표현해 주는 방법의 차이이다. 언어의 우열을 가려 사용하면 한 가지 언어만이 독보적으로 사용될 것이다. 그러나 특정 언어만이 갖는 강점이 있고, 다양한 개발 분야에서 특히 많이 사용되는 언어가 있다. 한 가지 언어를 깊이 알고 로직의 프로세서를 이해 할 수 있다면 다른 언어도 훨씬 빨리 배울 것이다.

4.3 비주얼 베이직 교육 내용 요소

비주얼 베이직의 기본 학습 내용 요소를 정리하면 표와 같다.

<표 3> 비주얼 베이직 기본 학습 내용 요소

구분	◆ 내용
프로젝트 Project	같은 일을 하는 관련된 파일들의 모임.
폼 Form	화면에 표시될 창(Window)
컨트롤 Control	폼 위에서 여러 가지 일을 하는 것.
개체 Object	모든 작업 단위나 대상. Form + Control
속성 Property	폼이나 컨트롤들의 작동과 특성을 결정하는 것.
이벤트 Event	프로그램 사용자나 시스템에 의해 어떤 동작이 취해지는 것.
모듈 Module	Visual Basic 코드로 구성된 단위.
프로시저 Procedure	일정한 기능을 수행하는 코드의 집합
함수 Function	매개변수를 전달하여 값을 얻어낼 수 있는 재사용할 수 있는 코드의 집합(block)
메서드 Method	특정 개체가 특정한 작동을 하게 하는 일종의 함수
연산자 Operator	하나 또는 그 이상의 값을 계산할 때 사용되는 용어
디버그 Debug	프로그램을 점검하거나 수정하는 일련의 작업
컴파일 Compile	프로그램 단독으로 실행할 수 있도록 실행형 파일을 만드는 작업

변수 Variables	일정 크기의 데이터를 저장하는 메모리의 주소값
상수 Constant	변화되지 않는 값(수나 문자, 문자열)
데이터 형 Data types	자체적으로 연산이 가능한 값(value)의 집합. 문자열(String), 정수형(Integer), 참. 거짓(Boolean)등
매개변수 Argument	함수에 값을 전달하기 위한 공간 또는 장소
배열 Array	같은 형의 많은 변수를 만들 때 사용. 정적 배열과 동적 배열
비교문	변수나 값, 상수의 연산을 통하여 진리의 값이 되면 어떠한 일을 수행하는 문장. If~Then~Else, Select Case
반복비교	비교문이 참일 경우까지 어떤 코드를 반복적으로 수행하는 문장. Do~Loop, For(Each)~Next
클래스 Class	새로운 형태의 오브젝트를 정의해서 만들어 내기 위한 데이터 형
생성자 Constructor	선언된 클래스의 형태를 가지고 있는 개체가 생성될 때 실행되는 메서드
소멸자	생성된 개체가 소멸될 때 실행되는 메서드
DAO	Database Access Object. 데이터 베이스 엔진에 의해 정의되어 있는 개체형

4.4 비주얼 베이직 교육 내용

비주얼 베이직의 기본 학습 내용 요소를 바탕으로 아래와 같은 지도 내용을 제안하고자 한다.

<표 4> 비주얼 베이직 프로그래밍 지도 내용

주제	학습 내용
규칙	• 프로그래밍 규칙 • 프로그래밍 용어 설명
개체	• 개체의 종류 • 개체의 속성 • 개체의 사용법
이벤트	• 이벤트 프로시저의 종류와 사용법 • 버튼 이벤트
디버깅과 컴파일	• 디버깅 • 컴파일과 배포본 만들기
변수	• 변수의 선언과 대입 • 변수의 값 유형
상수	• 상수의 종류 • 예약어
데이터형	• 데이터(data)의 종류 • 데이터 형 변환과 다루기
원시 데이터형	• 숫자형 • 문자열형과 문자열의 조작
연산자	• 연산자 우선 순위와 결합법칙 • 대입연산자(=) • 단항연산자
	• 산술연산자 • 관계연산자 • 기타연산자
구문	• 구문의 종류와 기본 문법 • 구문의 사용법
	• With~End With 문
	• If~Then 문 • If~Then~Else문 • If~Then~Elseif문

	<ul style="list-style-type: none"> • Select Case문 • 다중 조건문 • 반복문의 종류 • For~Next 문 • Do While~Loop 문 • Do~Loop While 문 • Do Until~Loop 문 • Do~Loop Until 문
함수	<ul style="list-style-type: none"> • 함수의 개념 • 함수 만들기 • 함수에 값 전달하기 • 함수 종료 및 값 리턴 • 함수사용 범위 • 함수 영역 • 함수 매개 변수 • 내부함수 • 사용자 정의 함수
메뉴 편집기	<ul style="list-style-type: none"> • 메뉴 편집기의 사용법 • 메뉴 편집기의 활용
배열	<ul style="list-style-type: none"> • 배열의 정의 • 배열의 구조 • 배열의 원소 • 배열 만들기 • 원소의 추가 • 원소의 삭제 • 다차원 배열 • 배열의 응용
파일 다루기	<ul style="list-style-type: none"> • 파일의 처리 • 순차 파일 • 랜덤 파일
DB 다루기	<ul style="list-style-type: none"> • 데이터베이스의 정의 • 데이터 다루기 • DAO • 데이터베이스 조작
모듈	<ul style="list-style-type: none"> • 모듈의 정의 • 모듈 사용법 • 모듈 활용하기
클래스	<ul style="list-style-type: none"> • 클래스 모듈의 개요
API 함수	<ul style="list-style-type: none"> • API 함수의 개요

6. 결론 및 제언

본 연구에서는 초등 정보과학영재를 위한 프로그래밍 교육의 필요성을 탐색하여 프로그래밍 교육내용 요소를 추출하였으며, 프로그래밍 언어 선정기준을 통해 비주얼 베이식을 교육용 프로그래밍 언어로 선정한 후 프로그래밍 교육내용을 제안하였다.

초등 정보 과학영재용 프로그래밍 교육과정 개발을 통해 다음과 같은 효과가 기대된다.

첫째, 각급 학교 및 기관에서 교육할 수 있는 프로그래밍 교육내용의 체계가 수립되고 지향점을 가진 초등 정보과학영재 교육이 이루어질 수 있을 것이다.

둘째, 프로그래밍에 대해서 전혀 알지 못하는 초등 정보과학영재일지라도 쉽게 프로그래밍의 기초를 확립할 수 있을 것이며 더불어 중등 진학 후 지속적으로 실시하게 될 상위단계의 프로그래밍 교육의 기초를 형성할 수 있을 것이다.

셋째, 비주얼 베이식의 쉬운 사용법과 다양한 기능은 아동의 창의성을 자극하고 발산하여 그들의 영재성을 개발하는데 도움을 줄 수 있을 것이다.

향후 계속해서 비주얼 베이식을 활용한 프로그래밍 교육 내용에 대한 연구가 필요

하며 학습자의 흥미와 수준에 맞는 예제의 개발, 학습한 프로그래밍적인 기능을 통합적으로 익히고 구현할 수 있는 다양한 프로젝트의 개발이 필요하다. 또한, 학습자의 특성을 반영할 수 있는 교수법에 대한 연구가 필요하다.

참고문헌

- 오승현, “영재교육, 주요 정책방향 및 향후 과제”, 교육부 [정책해설], 2002
- 나동섭, “초등 정보과학영재교육을 위한 교육과정의 개발”, 경인교대 석사학위 논문, 2003
- 나동섭, 이재호, “정보과학영재를 위한 교육분야 정의”, 한국정보교육학회, pp.378-379, 2001.
- 유철중, “초, 중등학교에서의 정보 영재를 위한 컴퓨터 프로그래밍 교육”, 전북교육 통권 제27호, pp.8-15, 2002
- 정두업 외 2인, “중학교 정보영재교육의 실태 및 개선방안”, 정보처리학회지, 제9권 제5호, pp.16-25, 2002
- 오상진, “애니메이션 저작도구를 이용한 초등 정보과학영재교육용 프로그래밍 교육과정의 개발”, 경인교대 석사학위 논문, 2003
- 이균현, “영재교육학”, pp.45-67, 1989.
- 이경화, “초등학생을 위한 로고 프로그래밍 지도방안”, 한국정보교육학회 2002년 하계 학술논문집, 제 7권, 제 2호, pp.303-304, 2002.
- 김홍환, “프로그래밍언어의 효과적인 교육을 위한 CAI 기법 연구”, 응용과학연구 제 10권 제 1호, pp.9-22, 2001.
- 이형배, “이형배의 비주얼 베이식 6”, 사이버 출판사, 1999
- 주경민 외 2인, “Visual Basic Programming Bible Ver 5”, 영진출판사, 1998
- Sally Fincher, What are we doing When we teach programming? In Frontiers in Education '99, pages 12a41 - 5. IEEE, November 1999.
- <http://www.seoul-e.ac.kr/guide/hakbo/last/327/7-1.html>