

# 초등학생의 논리적 사고력 및 문제 해결 능력 향상을 위한 컴퓨터 프로그래밍 교육과정 모델 제안

- 재량·특별활동시간에 비주얼베이직언어를 중심으로 -

문 외 식  
(진주교육대학교 컴퓨터교육과 교수)

---

## 《 요 약 》

---

7차교육과정에서의 초등 컴퓨터교육은 재량활동시간 또는 특별활동시간에 단순한 응용프로그램(워드프로세서, 엑셀, 파워포인트 등)을 활용하거나 인터넷을 이용한 정보검색 수준이다.

따라서, 문제해결 능력이나 논리적 사고력을 향상시키는데는 미흡하다. 인지능력이 뛰어난 초등학교 5, 6학년에는 이러한 컴퓨터 활용방법 교육보다 컴퓨터 기초원리 또는 프로그래밍 교육이 필요하다.

본 논문은 프로그래밍교육을 재량·특별활동 시간에 학년별(5, 6학년) 또는 통합학년으로 학습할 수 있게 프로그래밍교육과정 60차시 분을 개발하여 제안하였다. 개발한 교육과정은 교수 및 학습자의 요구사항을 최대한 반영하기 위해 설문조사 등을 통해 얻은 기초자료를 참조하였다.

제안한 교육과정이 초등학교 5, 6학년에 적합하고 우수함을 검토하기 위해 일부 차시의 교육과정으로 5, 6학년을 대상으로 재량활동시간에 직접 교수·학습한 후 자기평가, 설문조사를 실시하였다. 결과로 제안한 교육과정이 적합성과 우수성이 있다고 판단되었다.

주제어 : 재량특별활동시간, 초등학교 5, 6학년, 프로그래밍교육, 프로그래밍교육과정

---

## I. 서 론

현재 초등학교 컴퓨터교육의 대부분은 단순히 상업용으로 만들어진 응용프로그램(워드프로세서, 파워포인트, 엑셀 등)을 사용하거나 웹 브라우저를 통해 정보를 검색하는 수준으로 ICT학습의 본래의 목적인 문제해결 능력을 키우고 사고력을 증진시키는

데는 미흡하다. 컴퓨터교육에는 컴퓨터를 활용하는 교육과 컴퓨터 자체를 배우는 두 가지의 의미가 함께 포함되어 있어야 한다. 그러나 현재 컴퓨터교육의 기본골격은 가장 중요한 컴퓨터의 원리를 도외시 하고 지나치게 활용방법 즉, 단순 기능위주로 편성되어 있어 장기적으로는 컴퓨터를 활용하고 응용하는 능력을 떨어지게 할 수 있다.

선진정보강국으로 자처하는 미국, 일본에서는 활용위주의 컴퓨터교육과정을 개념과 원리를 기초로 하는 교육과정의 비중을 높여 학습자들의 문제해결능력 향상에 초점을 두고 있다. 특히, 신흥 정보강국으로 각광 받고있는 인도에서는 초등학교 2학년 때부터 로고나 베이직 등의 언어를 이용하여 컴퓨터의 기본원리, 프로그래밍원리를 먼저 배우고 고학년때 웹 및 파워포인트, 엑셀 등의 응용소프트웨어를 배우게 하여 어릴 때부터 흥미위주의 컴퓨터학습을 경계하고 창의성을 늘리는데 중점을 두고 있으며 크게 효과를 거두어 초등학교 때에 이미 IT인력양성을 위한 시발점으로 여기고 있다(정재열, 최재혁, 공영태, 2005). 우리나라는 이와는 대조적으로 7차교육과정 부터는 과거의 초등학교 교육과정에 있었던 컴퓨터 자체의 구조적 논리 이해와 논리적 사고력 및 문제해결 능력을 함께 키울 수 있는 프로그래밍 교육이 아예 빠져있다. 컴퓨터교육에 있어서 프로그래밍은 대단히 중요한 요소이지만 빠르게 발전하는 하드웨어 및 소프트웨어 기술로 인해 프로그래밍 능력보다 상업용으로 만들어진 응용프로그램을 활용하는 능력이 훨씬 중요하게 부각되었다. 따라서, 사용자의 관점인 편리성만 강조할 뿐 복잡하고 다양한 사고력을 요구하는 프로그램을 만드는 능력까지 굳이 초등학교에서 할 필요가 없다는 근시안적인 이유 등으로 프로그래밍 교육이 제외되어 있다.

그러나 프로그래밍 교육의 중요성은 프로그래밍 언어 습득 그 자체 보다 컴퓨터 프로그래밍 학습과정인 순서도 작성과 코딩을 통해서 다양한 논리적 사고력을 키울 수 있으며 오류 수정 등 스스로의 문제해결 과정을 통해서 반성적 사고와 문제해결 능력을 자연스럽게 키울 수 있다. 이러한 일련의 프로그래밍 학습경험은 과학 및 수학적 사고력을 향상시켜 초등교과의 전반적인 교과 학습능력을 높일 수 있다.

프로그래밍교육은 초등학생들이 배우기에는 다소 어려운 언어와 문법으로 구성되어 있어 가르치는 교수자나 배우는 학습자 모두에게 적지 않은 부담이 되고 있다. 또한, 현실적으로 볼 때 학교에서 프로그래밍교육은 많은 수업 시수와 전문 강사의 확보, 그리고 적절한 프로그래밍 도구의 선정 등 교육환경 및 전문성이 요구되는 여러 가지 장애물이 있는 것이 사실이다. 그럼에도 불구하고 프로그래밍 교육은 초등학교 전체 교과 학습향상에 많은 장점을 가지고 있기 때문에 장점을 흡수하면서 문제점을 피해갈 수 있는 적극적 방안을 모색하면 컴퓨터 활용을 위주로 하는 교육의 한계를 뛰어넘어 국가적 사업인 IT 및 과학기술 인력양성의 시발점을 초등학교로 앞당길 수 있는 획기적이고 발상의 전환점이 될 수 있다.

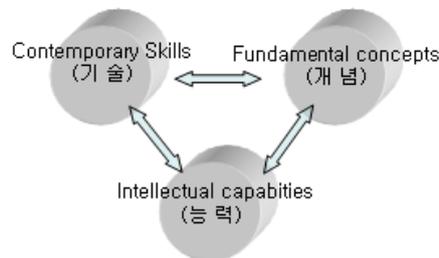
본 논문에서는 초등학생들이 비교적 쉽게 접근할 수 있고 호감이 갈 수 있는 비주얼 베이직(Visual Basic)을 도구로 선택하여 초등학교 고학년인 5, 6학년이 개별학년 또는 통합학년이 재량활동 또는 특별활동 시간에 학습할 수 있는 프로그래밍 교육과정 모델을 개발하였다. 개발한 교육과정 모델을 기초로 수업 및 지도계획서를 작성하여 재량·특별활동시간에 프로그래밍 교육을 실시한 후 학습자들의 흥미도, 학습능력 정도, 논리적사고력 및 문제해결 능력 향상에 어떤 영향을 미치는지 등 요인들의 결과를 분석하여 교육과정의 적절성을 검토하였다. 분석된 자료는 향후 업그레이드되는 최적 교육과정모형을 완성하는데 필요한 중요한 구성요소 자료로 활용할 수 있다.

## II. 관련연구

### 1. 컴퓨터교육

컴퓨터교육이란 일반적으로 응용소프트웨어 사용을 위한 소양교육, 컴퓨터를 활용하여 교과수업의 효과를 높이는 활용교육 그리고 컴퓨터개념 원리 및 프로그래밍학습을 통해 과학적 사고방식의 도입으로 창의성과 문제해결 능력을 높이는 교육이라 할 수 있으며 궁극적으로 초·중등 및 대학에 이르기 까지 다음과 같은 3가지 능력을 체계적으로 갖추게 하는데 목표를 두어야 한다(National Academy of Science, <http://www.nationalacademies.org/subjectindex/com.html>, 컴퓨터월드, 2005).

- ① 기술(Contemporary Skills) - 정보통신기술, 워드프로세서, 엑셀, 파워포인트 등 활용능력
- ② 개념(Fundamental Concepts) - 컴퓨터 개념 및 원리, 구조
- ③ 능력(Intellectual Capabilities) - 창의적인 문제해결 능력



[그림 1] 창의적인 컴퓨터(IT)교육의 3가지 요소

## 2. 초등학교 컴퓨터교육관련 교육과정

7차 초등교육과정은 초등학교 5, 6학년 실과과목의 일부분에서 소양교육을 할 수 있으며 또한, 국어, 영어 교과외의 작은 영역에서 소프트웨어 활용학습을 할 수 있는 영역이 일부 있다. <표 1>의 전체적인 내용을 살펴보면 현재의 초등학생들의 컴퓨터 수행능력을 제대로 반영하지 못한 것으로 대부분 이미 터득한 워드프로세서를 활용하는 간단한 수준으로 학습자들의 흥미도를 잃게하여 컴퓨터교육 자체가 무의미해 질 수도 있으며, IT(ICT)의 신기술 수명주기가 1년 정도라 본다면 현실성이 매우 결여되어 있다(문외식, 2002). 이러한 현실적 어려움을 해결할 수 있는 방법으로 재량활동시간 등을 통해 1 - 6학년까지 년 간 34시간을 적절히 활용함으로써 컴퓨터 소양 및 활용 능력을 크게 높일 수 있다. 컴퓨터를 다루는 소양 및 활용교육은 특성상 동시에 이루어지기가 어려우며 정규교과에 반영되어 있지 않기 때문에 독립적으로 교수·학습하기가 어렵다. 따라서, 재량활동시간에 1, 2, 3학년에는 소양교육 그리고 4, 5, 6학년에는 활용교육 중심과 기본적인 컴퓨터원리 및 프로그래밍 학습이 매우 효과적이다.

<표 1> 초등컴퓨터교육 관련 전체 교과·교육과정

과목	학년	영역	컴퓨터 관련 교육내용
국어	3	쓰기	<ul style="list-style-type: none"> <li>• 글을 컴퓨터에 옮겨 쓴다.</li> <li>- 기본: 글쇠판을 자신이 쓴 글을 컴퓨터에 옮겨 쓴다.</li> <li>- 심화: 글쇠판을 보지 않고 자신이 쓴 글을 컴퓨터로 옮겨 쓴다.</li> </ul>
	4	쓰기	<ul style="list-style-type: none"> <li>• 컴퓨터를 이용하여 자신의 생각을 글로 쓴다.</li> <li>- 기본: 컴퓨터를 이용하여 방학이 되면 하고 싶은 일을 글로 쓴다.</li> <li>- 심화: 컴퓨터를 이용하여 가족신문에 실을만한 내용을 글로 쓴다.</li> </ul>
	5	쓰기	<ul style="list-style-type: none"> <li>• 전달효과를 고려하여 자신의 글을 컴퓨터로 편집한다.</li> <li>- 기본: 글자모양, 크기, 문단모양 등을 고려하여 자신의 글을 보기 좋게 컴퓨터로 편집한다.</li> <li>- 심화: 그림, 사진, 표 등을 넣어 자신이 쓴 안내장이나 신문을 보기 좋게 편집한다.</li> </ul>
영어	4	읽기	<ul style="list-style-type: none"> <li>• 심화과정: 컴퓨터 자판의 알파벳 문자 익히기</li> </ul>
	5	읽기 쓰기	<ul style="list-style-type: none"> <li>• 심화과정: 컴퓨터 자판에서 알파벳 문자를 찾아서 치기</li> <li>• 심화과정: 컴퓨터 자판에서 구두로</li> </ul>
실과	5	생활 기술	<ul style="list-style-type: none"> <li>• 컴퓨터 다루기</li> <li>- 컴퓨터 구성</li> <li>- 자판 다루기와 글쓰기</li> </ul>
	6	생활 기술	<ul style="list-style-type: none"> <li>• 컴퓨터 활용하기</li> <li>- 컴퓨터로 그림 그리기</li> <li>- 컴퓨터 토인 활용하기</li> </ul>
재량 활동	1-6		<ul style="list-style-type: none"> <li>• 년 간 34시간 이상 활용 가능</li> <li>(1학년 30시간, 2-4학년 34시간, 5,6학년 실과교과/특별활동 활용포함)</li> <li>※초.중등학교정보통신기술운영지침(2000년)</li> </ul>

## 가. 프로그래밍교육

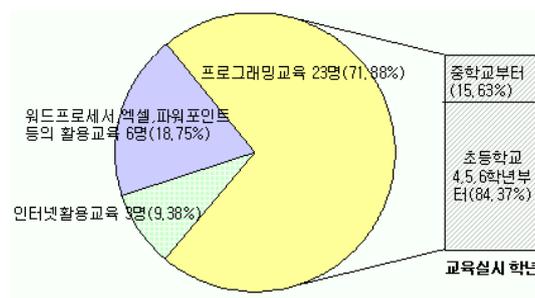
### 1) 프로그래밍교육이 미치는 교육적 효과

프로그래밍교육이란 문제해결을 알고리즘으로 설계 및 구현하여 이를 프로그래밍언어라는 도구로 표현할 수 있는 복합적인 능력을 배양하는 교육을 말하며 전통적인 교과에 미치는 교육적 효과는 다음과 같다(이경화, 김갑수, 2002).

첫째로, 프로그래밍과정을 통해 반복되는 오류의 유형을 쉽게 분석하고 스스로 사고하고, 문제를 해결할 수 있는 자기 반성적이고 논리적인 능력을 키울 수 있어 전통적 교과 학습을 자기주도적으로 이끌어 나갈 수 있는 능력을 배양하는 유용한 교육적 도구가 될 수 있다. 둘째로, 프로그래밍 학습은 자연적으로 응용소프트웨어가 컴퓨터 내부에서 동작하는 과학적인 원리를 이해할 수 있어 하드웨어의 추상적 개념과 과학·수학적인 개념을 자연스럽게 배울 수 있다. 셋째로, 단순한 활용위주의 교육에서 탈피하여 미래 세계경쟁력을 한 단계 높일 수 있는 IT기술 능력 교육을 초등학교에서부터 시작할 수 있는 계기를 마련하고 정보과학의 영재성을 조기에 발굴할 수 있어 효과의 정도에 따라 국가수준의 컴퓨터교육의 발상과 기본방향을 재 정립할 수 있다.

### 2) 논리적 사고력과 문제 해결 능력을 향상시키는 컴퓨터교육 교과와 교육 시기 조사

컴퓨터교육전공 석사학위를 소지하고 경남지역 각 초·중등학교에서 정보교육을 담당하고 있는 32명의 교사들에게 설문조사한 결과 [그림 2]와 같이 나타났다.



[그림 2] 논리적 사고력, 문제 해결력을 향상 시키는 컴퓨터교육과 교육시기

컴퓨터교육 중 프로그래밍교육이 초등학생들의 논리적 사고력과 문제해결 능력을 향상 시킬 수 있는 최적의 교육적도구라 생각(23명: 71.88%)하고 교육의 시기를 인지

능력이 활발한 4, 5, 6학년때 부터 시작하는 것이 바람직하다고 응답한 교사는 23명중 84.37%로 나타났으며 고등학교 또는 대학부터 프로그래밍교육이 시작되어야 한다고 응답한 교사는 없었다.

### 3) 프로그래밍교육의 교수-학습방법

컴퓨터를 이용하는 교수·학습 형태는 개인교수형(tutor), 보조도구형(tool), 학생주도형(tutee)으로 크게 분류할 수 있다(허희옥외, 2001).

개인교수형은 전통적인 교사의 역할을 컴퓨터가 대신하는 형태로 대부분의 CAI 소프트웨어가 이러한 형태에 속하며 현재 각광을 받고 있는 E-learning 또는 U-learning 형식을 띄고 있는 학습 방법도 이 형태에 속할 수 있다. 보조도구형은 결과를 얻기 위해 컴퓨터를 도구로서 단순히 사용하는 경우로 흔히 우리가 사용하는 응용소프트웨어(워드프로세서, 스프레드시트 등)를 활용하는 형태를 말한다. 학생주도형은 학생 자신이 직접 구성된 논리적 프로그램 과정을 통해 컴퓨터를 조작하여 실행하도록 하여 결과를 얻고 스스로의 사고과정을 통해 반성하는 발견적 교수방법 형태로서 BASIC 등의 프로그래밍언어를 이용하여 과정과 결과를 얻는 형태가 될 수 있다.

피아제는초등학생들의 지적인 성장은 교사의 지도에 의해서 이루어 지는 것이 아니라 아동 자신의 경험이나 사회적 상호작용, 성숙 및 평형 등에 의해서 크게 영향을 받는다고 한다. 따라서, 학생 자기주도형의 프로그래밍교육은 초등학생 자신들이 하고 있는 프로그래밍 활동이 가치있고 의미있는 일이라는 것을 스스로 느끼도록 학습환경을 만들 필요가 있다

프로그래밍 학습 시 발생하는 경험적인 문제점들을 정확한 데이터로 제시하기에는 어려우나 일반적으로 목표설정 달성에 미달하는 큰 요인을 두 가지로 정리할 수 있다. 첫째, 기능 및 문법 중심으로 교수.학습이 진행되기 때문에 학습시간과 많은 사고력이 소요되는 알고리즘 구현에 소요되는 충분한 시간 배정이 어려워 완성도 있는 프로그래밍 작성에 어려움이 있다. 둘째, 학습자의 개별수준을 무시한 교수.학습 방법으로 학습 중반 이후 학습자의 수업 참여도가 현저히 떨어지는 경향이 있다(김미량, 2002).

인지능력이 중등에 비해 비교적 떨어지는 초등학생을 위한 프로그래밍 교수·학습 환경을 만들기 위해서는 먼저, 학습자들은 자신의 수준에 맞는 능동적인 프로그래밍 활동을 하고 교수자는 모니터만 해 주고 가능한 손길이 적게 가도록 하는 가변적인 교육과정을 만들 필요가 있다. 즉, 기존의 교육과정과는 특별히 다른 느슨한 교수과정 이거나 순차적 교육과정이 아닌 수준에 맞게 선택하여 학습할 수 있는 학습 환경을 만들어야 된다.

교수·학습방법의 유형은 프로그램언어를 처음 배우는 초등학생들에게 언어의 이론적 학습을 위해서는 개인교수형, 그리고 실습을 위해서는 학생주도형과 발견적 교수방법을 함께 적용한 느슨한 교수·학습 방법을 함께 고려한 가변적이고 수준별 교육과정과 교재개발이 필요하다.

## 나. 프로그래밍교육 교과영역

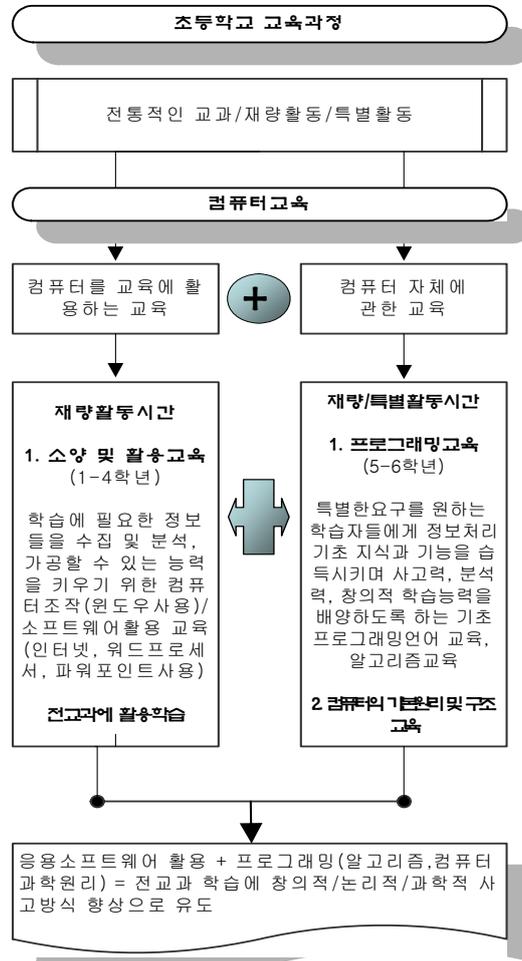
### 1) 전통적인 교과와 재량·특별활동

초등학교의 교육과정은 전통적인 교과와 특별활동의 두 영역으로 편성되어 있으며 여기에 재량활동을 신설하여 교육과정의 편제를 병렬적으로 3대 영역으로 구성되어 있다(교육인적자원부, 2005). 전통적인 교과는 학문적이고 인지적인 필요와 요구로부터 만들어져 학습하지 않으면 안되는 미리 준비되고 고정된 형태이다.

그러나 특별활동은 학습자의 인간적, 사회적 필요와 요구로부터 만들어져 있다. 따라서, 학습자의 필요와 선택, 개성과 형편에 따라 학생이 주체가 되어 융통성 있게 운영할 수 있는 유연한 구조의 교육과정이라 볼 수 있다. 재량활동은 미래사회를 주도적으로 이끌어 갈 수 있는 자기주도적인 능력과 창의성을 신장시키기 위해 학교나 지역사회실정, 교원, 학생, 학부모의 필요와 요구를 반영하고 학교의 교육적 환경에 알맞게 창의적이고 특색있는 운영을 함으로써 궁극적인 목표인 인간교육을 실현하는데 있다.

초등학교에서의 교육과정은 교과, 재량활동, 특별활동을 별개의 영역으로 구분하지 않고 이들 3교과가 서로 유기적으로 관련지어져야 한다. 초등학교에서의 컴퓨터교육의 최적 영역으로는 재량활동·특별활동이라 볼 수 있으며 이는 본래의 국가적 목표에 맞게 운영할 수 있는 영역이라 할 수 있다.

따라서, 전통적 교과교육에 활용할 수 있는 ICT소양 및 활용교육은 저학년(1-4학년)들을 대상으로 재량활동시간을 활용하고, 사고력과 인지능력이 가장 뛰어난 5, 6학년을 대상으로 재량활동·특별활동시간에 흥미, 소질, 적성이 비슷한 학생들을 자발적으로 통합 참여시켜 잠재능력과 창의성을 개발시킬 수 있는 기초 프로그래밍 및 알고리즘 교육이 적절하다.

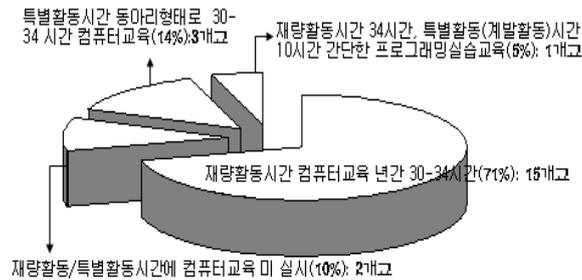


[그림 3] 전통적교과, 재량·특별활동, 컴퓨터교육과의 관련구조

## 2) 재량·특별활동시간에 컴퓨터교육 실시 현황

무작위하게 선택한 경남지역 소재 21개 초등학교의 재량·특별활동 교육과정 운영 현황 중 5, 6학년의 재량·특별활동시간의 교육과정 편제 및 시간배당을 분석하면 컴퓨터교육과 창의적 재량활동이란 2개의 큰 영역으로 대부분 ICT 활용교육과 한자, 주제탐험, 인성교육 등을 편성하여 각각 년 간 34시간 정도 활용하고 있다. [그림 4]는 경남지역 21개 초등학교에서 실시 운영되고 있는 재량·특별활동시간에 편성되어 활용되는 컴퓨터교육 시간을 나타내고 있다. 조사대상 학교의 71%인 15개교는 재량활동 시간에 컴퓨터에 관련된 교육과정을 편성하여 실제로 교육을 실시하고 있는 것으로

나타났으며 21개교 중 1개교가 특별활동시간에 10시간을 프로그래밍교육에 사용하고 있는 것으로 조사되었다.



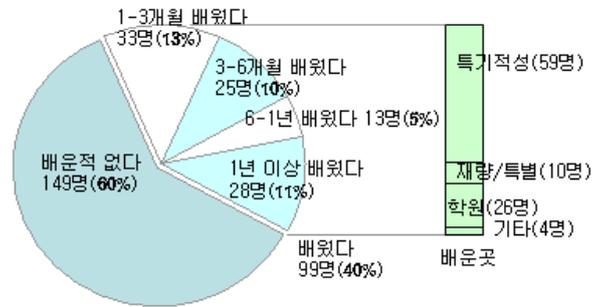
[그림 4] 5, 6학년 재량·특별활동시간 컴퓨터교육 실시 현황

#### 다. 프로그래밍 학습 경험 조사 및 교수자 선호도

##### 1) 프로그래밍 학습 경험 조사 및 분석

초등학생 5, 6학년을 대상으로 재량·특별활동시간용 프로그래밍 교육과정 개발을 하기 위해 5, 6학년들의 프로그래밍 학습 경험을 설문조사하였다. 설문조사 시기는 2005년 5월에 실시하였으며 경남의 도시지역 학교 6개교, 농·어촌 지역 10개교 총 248명의 5, 6학년생을 대상으로 하였으며 설문대상의 전체 학생 중 컴퓨터를 보유하고 있는 학생이 242명 그리고 보유하고 있지 않는 학생이 6명으로 설문대상의 학생 중 98%가 개인용 컴퓨터를 보유하고 있어 대부분의 학생들이 어디서나 컴퓨터 프로그래밍 학습이 가능한 환경을 가지고 있었으며 다음과 같이 분석되었다. 설문대상의 전체학생 248명 중 프로그래밍을 배운적이 없는 학생들이 149명으로 전체 60%를 차지하고 99명인 40%가 프로그래밍 학습을 경험한 것으로 비교적 높게 나타났다. 이는 일부 응답자가 프로그래밍과 응용소프트웨어의 개념을 혼돈하여 배운 프로그래밍이 파워포인트, 엑셀 등이라 응답한 학생들이 있어 실제 프로그램 학습자는 다소 적을 것으로 판단된다. 프로그래밍을 경험한 학생 중 학교에서 재량·특별활동시간에 프로그래밍을 배웠다는 학생은 10명이며 프로그래밍을 경험한 나머지 90% 이상인 89명의 학생은 사교비가 드는 특기적성시간, 학원 등지에서 배우고 있는 것으로 분석되었다.

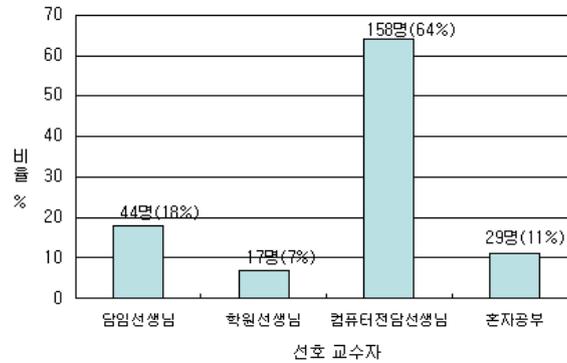
따라서, 사교육비 경감 차원에서도 초등학생들의 프로그래밍 학습 욕구를 학교내의 교과과정(재량·특별활동시간)에 반영하는 것을 적극 검토할 필요가 있다.



[그림 5] 초등 5, 6학년생들의 프로그래밍학습 경험과 배운 곳

## 2) 프로그래밍 교수자 선호도조사

초등학생 5, 6학년들의 248명중 프로그래밍교육을 받는다면 누구에게 받는 것이 가장 좋겠는가에 대한 설문조사 결과는 [그림 6]과 같다. 응답자 중 155명인 64%가 컴퓨터 전담선생님을 선호하고 있으며 담임선생님도 응답자의 18%인 44명으로 프로그래밍 교수자로서 현직의 선생님을 선호하고 있는 것으로 나타났다.



[그림 6] 프로그래밍 교수자 선호도

## 라. 교육용 프로그래밍 언어와 비주얼베이직

### 1) 교육용 프로그래밍 언어

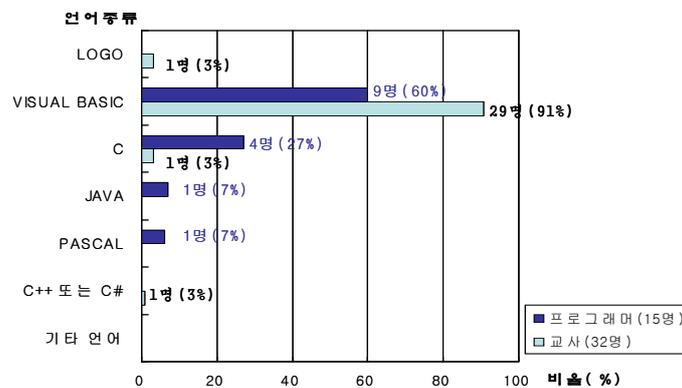
현재 아주 많이 널리있는 소프트웨어(워드프로세서, 엑셀, 컴퓨터게임 등)들은 내부

에 질서정연한 수많은 코드들로 이루어져 있다. 사용자 자신만의 목적이나 주변 사람들의 공통된 목적에 맞는 응용프로그램을 만들어 사용하려면 프로그램언어로 응용프로그램 코드를 작성해야 한다. 이러한 소프트웨어 개발을 목적으로 사람들은 프로그래밍 언어를 배운다. 그러나 초등학교에서의 프로그래밍 교육은 학습자들에게 새로운 교과내용으로서 색다른 교육의 경험을 얻게하고 풍부한 논리적 원리 및 추론 경험을 배워서 전통적인 교과 학습력 신장에 도움을 주고 사고력 및 문제해결 능력 신장에 초점을 두어야 한다. 이러한 목적을 달성하기 위해서는 초등학교 교육용 프로그램언어의 선택요건은 다음과 같은 조건이 필수적이다(강성원,이애정,이재호, 2003).

- ① 문법과 표현이 비교적 간단하여 코드작성과 오류수정이 용이해야 한다.
- ② 프로그램 작성과 결과를 쉽게 인식할 수 있게 시각적으로 볼 수 있는 비주얼 환경 언어이어야 하며 또한, 쉽게 다른 프로그램언어로 전환이 용이한 코드호환 및 문법의 대중성이 있는 것이라야 한다.

## 2) 교육용 프로그래밍 언어 선호도 조사

2005년 3월 경남지역 32명의 초·중등 정보담당교사 그리고 IT산업 현장에서 종사하고 있는 15명의 프로그래머들에게 초등학교생들에게 프로그래밍교육을 시킨다면 가장 적합한 언어가 무엇인가를 설문조사한 결과 [그림 7]과 같이 나타났다.



[그림 7] 교사·프로그래머의 교육용 프로그래밍 언어 선호도

[그림 7]의 결과에서 비주얼베이직이 가장 적합하다고 응답한 비율이 교사와 프로그래머

그래머가 각각 91%와 60%로 나타났다. 그리고 IT산업종사 프로그래머들에게는 일부이지만 교사가 별 반응이 없거나 전혀 응답하지 않았던 C, JAVA, PASCAL도 초등학교 교육용으로 유익하다고 설문에 응답하고 있다. 이는 현장에서 많이 사용되는 프로그램의 인지도가 반영된 것으로 판단되나 교사의 입장에서는 초등학생이 수용 가능한지가 변수로 작용한 것으로 판단된다.

### 3) 비주얼 베이직 언어

초등학교 교육용으로 적합한 프로그래밍언어로서는 BASIC, PASCAL, C, LOGO 등을 들 수 있다. 이중에서도 초등학생들에게 보다 쉽고 재미있게 문제를 해결하도록 고안된 LOGO 및 BASIC 프로그래밍언어는 과거 컴퓨터교육에서 프로그래밍언어를 중시하던 시절부터 널리 사용되어 왔다. LOGO는 1960년대 후반 Seymour Papert가 아동의 학습을 위해 고안된 프로그래밍언어로 학습자가 다양한 수학과제를 재미있게 해결하기 위한 고유의 명령어와 오류수정 등의 편리성이 있어 초등학교에서 자주 사용되어 왔다. 그러나 학습자들인 초등학생들이 앞으로 배워서 사용하게 될 다른 컴퓨터 언어 구조를 이해하고 배우는데는 BASIC, C, PASCAL에 비해 호환성이 많이 떨어지며 지속적인 LOGO언어의 업그레이드 면에서도 뒤 떨어진다. 또한, C 또는 PASCAL 언어 경우 역시 초등학생들이 작업하기 좋은 비주얼 환경성이 떨어지며 문법 및 명령어들을 초등학생들이 소화해 내기에는 다소 무리가 있다.

초등학생들의 흥미를 유발하고 이해가 보다 쉬운 그래픽 환경이 지원되는 프로그래밍 언어로 비주얼베이직을 들 수 있다. 비주얼베이직은 1991년 미국의 마이크로소프트사에서 발표되었으며 처음 프로그래밍언어에 입문하려는 초보자들에게 적합한 언어로 예전부터 사용되어 오던 기존의 BASIC 코드를 그대로 전수받아 흔히 보았던 코드들이라 프로그래밍을 교육시키는 비전공자 교수자들인 교사에게는 보다 더 친숙한 언어이기도 하여 교사 자신들의 재교육에도 효과적이다.

교육용으로 사용되어온 기존의 많은 프로그램 언어들은 사용자와 컴퓨터간의 인터페이스를 위해 직접 많은 줄의 명령코드를 코딩하였으나 비주얼베이직은 인터페이스 각각의 구성요소를 미리 만들어진 개체를 이용하여 화면에다 끌어놓으면 자동으로 생성되는 우수한 환경인 GUI방식이며 학습자인 초등학생들이 향후 다른 프로그램으로 전환하여 공부하는데 용이하며 지속적인 업그레이드로 성인이 된 후에도 계속 사용할 수 있다. 본 논문에서는 비주얼베이직의 여러 버전 중 초등학생들에게 가장 적합한 버전 6.0(학습판)을 선택하였다.

### Ⅲ. 프로그래밍 교육과정 모델 제안

Ⅱ장의 관련연구에서 조사하고 분석된 자료를 기초로 학습자·교수자의 요구도를 반영한 프로그래밍교육과정을 개발하여 제안하였다. 5, 6학년이 재량·특별활동시간에 학급별 또는 학년별 개별·통합수업에 활용할 수 있도록 교육과정을 모두 60차시로 나누고 영역, 세부영역 학습주제 그리고 학습주제에 대한 학습내용과 학습관계를 가능한 자세하게 기술하였으며 제안한 교육과정을 단계별로 쉽게 교재로 개발하도록 구성하였다(문외식, 1999; 김웅석, 1999; Allen Tucker et al, 2003; Computer Curricula, 2004).

<표 2> 프로그래밍 교육과정

차시/분	영역	세부영역 학습주제	학습 내용	학습 단계
1-2/60	컴퓨터와 프로그래밍 언어	컴퓨터의 기본구성과 프로그래밍 언어 개요	1. 컴퓨터는 어떻게 이루어져 있을까?(하드웨어/소프트웨어 개요 설명) 2. 프로그래머와 프로그래밍언어(고급언어/기계어/컴파일러 개요설명) 3. 컴퓨터에서 프로그램을 처리하는 과정	학습 준비 단계
3/60		비주얼 프로그래밍 컴파일러 설치하기	1. 비주얼베이직을 내컴퓨터에 설치하기(따라하기) ※ 평가판/학습판 제공 및 다운로드	
4-5/60	비주얼 베이직 실행하기	비주얼베이직 실행하기 및 전체모습 살펴보기	1. 비주얼베이직 실행 따라하기 2. 전체모습(비주얼베이직 인터페이스) 살펴보기 (폼/도구모음/프로젝트 탐색기/도구상자/속성창 기능 설명)	기본 단계
6/60		기본 구성 요소 실행하여 확인하기	1. 도구상자 기본기능(텍스트박스/커멘트버튼) 2. 속성 창 기본기능(Name/Caption) 3. 위의 1,2를 이용하여 화면에서 직접 실험해 보기	
7-10/60	컨트롤 속성 이해하기	커멘트 버튼 및 텍스트박스 속성, 포커스 이해하기	1. 커멘트버튼의 기능과 속성((이름)/Caption/enable/Visible/Default)들의 기본 기능 설명하기 2. 텍스트박스의 기능과 속성((이름)/Caption/Locked/MultiLine/Text1)들의 기본기능을 이해시킨다.	기본 요소 이해 단계
		레이블과 스크롤 바 속성 이해하기	1. 레이블의 기능과 속성((이름)/Alignment/AutoSize)들의 기본기능을 이해시킨다 2. 스크롤바의 기능과 속성(Value/Min/Max)들의 기본기능을 이해시킨다.	
		프로그래밍 하기; 간단한 워드프로세서 만들기와 프로그램 저장 - 프로그램 작성과정과 실행 및 저장과정 -	1. 텍스트박스와 스크롤바 두 가지만을 이용하여 간단한 워드프로세서를 만든 후 저장하고 실행한다. 2. 프로그램 작성과정과 실행 및 저장과정을 반복하여 학습시킨다.	
11-12/60		옵션 및 체크박스 속성이해하기	1. 옵션 및 체크박스의 기능과 속성(Caption/Value)	

13-16/60			들의 기본 기능을 이해시킨다.	
		프로그래밍 하기 <sub>2</sub> 옵션 체크박스 이용 주문형 식단표 만들기	1. 레이블, 커맨드버튼, 텍스트박스, 체크박스를 이용한 단순한 프로그램을 작성하고 실행 및 저장하기	
		프레임 속성 및 리스트 속성이해하기	1. 프레임 기능과 속성(Caption/Enable/Visible)들을 이해시킨다. 2. 리스트박스 기능과 속성(List/Multiselect)들을 이해시킨다.	
		프로그래밍 하기 <sub>3</sub> 프레임, 리스트 박스로 비상연락망 만들기	1. 커맨드버튼, 레이블, 텍스트박스, 리스트박스를 이용하여 간단한 연락용 주소 입력 프로그램을 만들고 실행한 후 저장해 본다. 2. 프로그램 작성과정과 실행 및 오류확인, 수정 그리고 저장과정을 반복하여 학습시킨다.	
		타이머 컨트롤 속성 알아보기	1. 타이머컨트롤의 기능과 속성(Enable/Interval)을 이해시킨다.	
	프로그래밍 하기 <sub>4</sub> 전자시계 만들기	1. 레이블, 타이머, 커맨드버튼을 이용하여 시계를 만드는 프로그램 작성하기 2. 프로그램 작성과정과 실행 및 오류확인, 수정 그리고 저장과정을 반복하여 학습시킨다.		
17-19/60		코딩규칙 및 변수이해하기, 변수선언 하기	1. 프로그래밍 시 코딩이 무엇이며 왜 하는가를 설명한다. 2. 기본적으로 코딩하는 방법(한 문장을 여러 줄로 나누기, 명령어 연결하기, 설명달기)을 설명한다. 3. 변수가 무엇이며 왜 사용하는지를 설명한다. 4. 변수를 사용할 때 필요한 규칙 몇 가지를 설명한다. 5. 코딩 시 변수를 선언하는 이유를 예로 설명한다.	
	변수/데이터 형이해하기	프로그램에서 사용되는 데이터 기본유형 알아보기	1. 프로그램에서 왜 데이터형을 선택하는지를 설명한다. 2. 기본 데이터형인 Integer, Long, Single, String에 대해서 설명한다.	
20/60		프로그래밍 하기 <sub>5</sub> 데이터형과 변수를 이용한 뺄셈 계산기 만들기	1. 커맨드버튼/텍스트버튼과 그리고 Integer 데이터형을 이용하여 간단한 뺄셈 계산기 만들기 2. 프로그램 작성과정과 오류수정 그리고 저장방법을 숙지시킨다.	프로그램 구조 이해 단계
21/60		프로그래밍 하기 <sub>6</sub> 데이터형을 이용한 메시지 전달기 만들기	1. 커맨드버튼/텍스트버튼과 String 데이터형을 이용하여 입력한 메시지를 출력하는 간단한 프로그래밍 하기 2. 프로그램 작성과정과 오류수정 그리고 저장방법을 숙지시킨다.	
22-24/60	연산자 이해하기	산술, 비교, 논리 연산자 이해하기	1. 산술연산자(덧셈/뺄셈/곱셈/나눗셈), 비교연산자(크다/작다/같다/같지 않다/작거나 크거나 같다), 논리연산자(논리곱/논리합) 사용에 대해 설명한다.	
		프로그래밍 하기 <sub>7</sub> 연산자를 이용 프로그래밍 코딩하여 확인하기	1. 산술연산자를 이용한 간단한 프로그래밍하기 2. 비교연산자를 이용한 간단한 프로그래밍하기 3. 논리연산자를 이용한 간단한 프로그래밍하기	
25-26/60		If문 이해하기	1. 프로그래밍 시 사용되는 기본제어구조를 설명하고 If구조를 설명하고 이해시킨다.(If 조건 Then 문	

			장/If 조건 Then 문장1 Else 문장2)	
		프로그래밍 하기 <sub>8</sub> If문 응용, 조건에 따른 간단 성적 처리 프로그램 만들기	1. If 조건 Then 문장1 Else 문장2 구조를 이용하여 점수를 텍스트박스에 입력하면 “잘함”, “보통”, “못함”을 출력하는 프로그래밍 하기 2. 프로그램 작성과정과 오류수정 그리고 저장방법을 숙지시킨다.	
27-28/60	제어 구조 이해하기	For - next문 이해하기	1. For -next 제어구조를 설명하고 간단한 예를 만들어(1에서 100까지 합계) 이해시킨다.	
		프로그래밍 하기 <sub>9</sub> For - next문을 활용하여 덧셈계산기 만들기	1. 1에서 100까지의 수 중 홀수들의 합만을 계산할 수 있는 알고리즘 및 플로우차트 함께 만들고 발표해 보기 2. 커멘드버튼과 텍스트버튼을 함께 사용하여 1에서 100까지 수 중 홀수의 합계를 계산하는 프로그래밍 하기	
29/60		Goto문 이해하기	1. Goto문 구조 및 사용방법에 대해 설명하고 간단한 예제 프로그래밍을 이용하여 이해시킨다.	
30-34/60	배열 이해하기	배열 이해하기 1차원 배열	1. 배열이란 무엇인가를 그림 등을 이용하여 기본 개념을 설명하고 이해시킨다. 2. 1차원 배열을 이용하여 간단한 프로그래밍 하기 (각 배열에 숫자를 넣고 이를 출력해 보는 프로그래밍 하기)	
		2차원 배열	1. 2차원 배열의 기본구조 및 개념, 사용형식을 그림 등으로 설명하고 간단한 프로그래밍하기(위의 1차원 배열에서 작성한 프로그래밍을 2차원 배열로 고쳐서 실행해 보기)	
35-37/60	프로시저 이해하기	함수프로시저 이해하기	1. 프로그래밍에 사용되는 함수와 프로시저의 기본 개념을 설명하고 이해시킨다.	
		프로그래밍 하기 <sub>10</sub> 함수프로시저를 이용하여 사칙계산기 만들기	1. 함수/프로시저를 이용하여 덧셈/뺄셈/나눗셈/곱셈을 할 수 있는 계산기 프로그래밍 하기 2. 프로그램 작성과정과 오류수정 그리고 저장방법을 숙지시킨다	
38-42/60	프로그램 메뉴 및 도구모음 만들기	메뉴 만들기의 이해	1. 비주얼한 프로그래밍을 하기 위해 사용되는 기존 메뉴들을 예와 함께 설명한다. 2. 프로그램에서 메뉴를 만들어 넣는 기본적인 방법을 설명하고 이해시킨다.	
		메뉴 만들기 실습하기	1. 메뉴(열기/저장/끝내기)가 만들어진 메뉴를 만들어 사용하는 간단한 워드프로세서 만들기를 프로그래밍 한다. 2. 1. 의 메뉴를 삽입하는 프로그램을 가능한 반복해 작성하도록 시킨다.	
43-47/60		도구모음 이해하기	1. 프로그램에서 자주 사용되는 도구모음의 기능과 편리성을 설명하고 이해시킨다. 2. 프로그램에 도구모음을 삽입하는 기본적인 방법을 설명하고 이해시킨다.	
		도구 모음 실습하기	1. 위의 메뉴 만들기에서 실습한 프로그래밍에 열기	

			와 저장하기 도구모음을 삽입하는 프로그래밍 해 보기 2. 반복하여 작성시켜 본다.	
48-52/60		설문조사 프로그램 만들기1 - 만들 내용 구성하기	1. 2-3명이 한조가 되어 협동 프로젝트 학습 2. 만들 내용을 각각 구성하여 그림으로 그리고 플로우차트 작성하기(각조는 다른 모양의 프로그래밍 하기) 3. 프로그래밍하기 와 오류수정하기 및 실행하기 4. 저장하기 5. 작성한 프로그램 발표하기	프로그램, 알고리즘 이해 단계 (협동학습)
		설문조사 프로그램 만들기2 - 품창 디자인하기/플로우차트 그리기		
		설문조사 프로그램 만들기3 - 관련요소 코딩 및 디버깅하기		
53-57/60	프로그래밍하기	구구단 프로그램 만들어 보기1 - 만들 모습 구성하기 및 방법 탐색하기	1. 2-3명이 한조가 되어 협동 프로젝트 학습 2. 만들 내용을 각각 구성하여 그림으로 그리고 플로우차트 작성하기(각조는 다른 모양의 프로그래밍 하기) 3. 프로그래밍하기 와 오류수정하기 및 실행하기 4. 저장하기 5. 작성한 프로그램 발표하기	
53-57/60		구구단 프로그램 만들어 보기2 - 만들 품창 디자인하기/플로우차트 그리기		
		구구단 프로그램 만들어 보기3 - 각 요소에 대하여 코딩하기		
		구구단 프로그램 만들어 보기4 - 테스트 및 디버깅하기		
58-60/60	평가, 반성하기	평가 및 반성하기 - 잘 공부했나요? - 프로그램을 얼마나 이해 했나요? - 프로그래밍 시 가장 어려웠던 문제점은 무엇 인가요? - 프로그래밍 시 어떤 오류가 가장 많이 발생 했나요?	1. 평가는 조사지 또는 웹설문조사를 통해 분석과 반성	평가 및 반성 단계

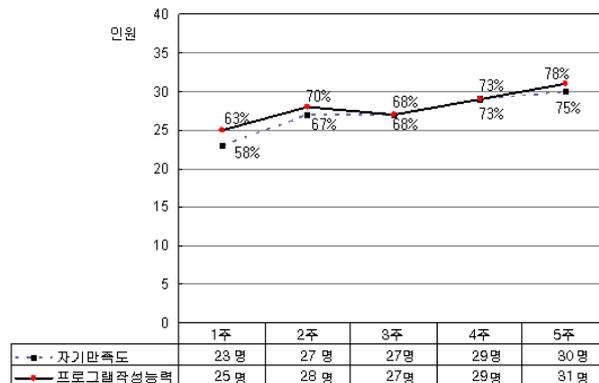
#### IV. 교육과정 모델 평가

제안한 프로그램 교육과정의 일부인 1차시에서 10차시까지의 내용을 교재로 작성하여 경남 A교육대학교 부설초등학교 5, 6학년 각 20명씩을 통합하여 재량활동시간에 2005년 6월 3째 주부터 7월 3째 주까지 5주간 주당 2시간씩 프로그래밍교육을 실시한 후 학습자의 자기평가 및 설문조사 결과 제안한 교육과정의 기본적 평가를 다음과 같이 얻었다.

##### 1. 학습자 자기 만족도와 프로그램 작성 능력의 변화

[그림 8]은 세부영역 학습주제별 학습 종료 후 프로그램 완성도, 학습과정 만족도를

자기 평가기준표에 의해 자기평가 결과(만족한다)를 주별로 누적한 결과와 학습 후 각 주별 1개씩의 프로그래밍 과정을 통해 프로그램을 완성한 학습자 수를 나타내고 있다. [그림 8]에서 볼 수 있듯이 학습자 자신의 자기만족도와 프로그램 작성능력은 각각 같은 비율로 조금씩 증가하는 것을 볼 수 있다. 주별 프로그램 완성율이 조금씩 늘어나고 있지만 학습내용의 난이도가 높아지고 사용해야할 도구들이 점점 많아진 점을 감안하면 프로그래밍 능력이 상당히 향상되고 있음을 알 수 있다.



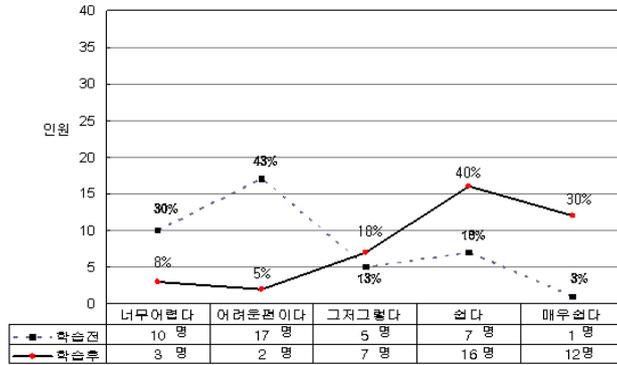
[그림 8] 자기만족도와 프로그램 작성능력 변화

## 2. 프로그래밍 학습난이도와 프로그램 학습 욕구의 변화

교육과정에서 제안한 학습내용의 난이도를 평가해 보고 학습이 진행 한 후 학습자들의 프로그램 학습 욕구가 어떻게 변화하였는지를 학습 전·후의 설문평가에 의해 분석하였다.

### 가. 프로그래밍 학습난이도에 대한 생각 변화

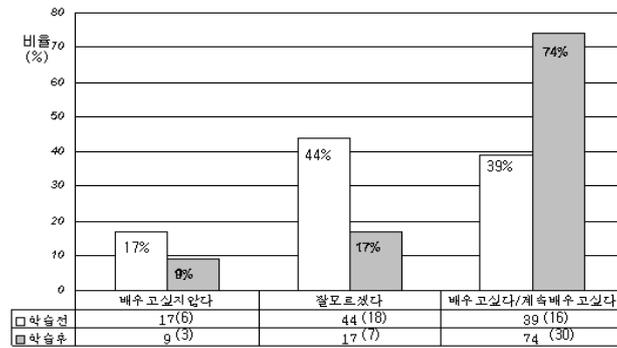
단위시간별로 학습주제 및 학습내용이 달라 검증할 영역별로 정확한 변화정도를 분석하기에는 다소 어려움이 있었으나 프로그램을 학습 전에 미리 알고리즘 형태로 구성하여 학습케 한 결과 73% 이상 대체로 어렵다고 응답하였으나 프로그래밍 학습 후 프로그래밍 작성과정 과 결과에서는 70% 정도가 대체로 쉽다고 응답을 하였다. 이는 프로그래밍 학습 후 점차로 사고의 능력이 향상되고 있다고 분석되며 대체로 프로그래밍 교육과정이 5, 6학년생들에게 적절한 수준으로 구성되어 있는 것으로 판단된다.



[그림 9] 학습 전·후의 프로그래밍 학습 난이도

#### 나. 프로그램 학습욕구 변화

프로그램 학습 전 단계에서는 배우고 싶지 않다고와 잘 모르겠다는 부정적 또는 일부 부정적인 응답을 한 학습자가 전체의 61% 정도 였으나 10차시 까지의 학습을 통해 계속 배우고 싶다고 응답한 학습자가 전체 응답자 중 74%로 늘어나 학습전의 39%에 비해 무려 35% 증가해 프로그램 학습의 지속력 향상과 프로그래밍에 대한 흥미가 높아 졌음을 알 수 있다.



[그림 10] 학습 전·후의 프로그램 학습 욕구

## V. 결론 및 향후연구

2001년부터 초등학교 재량활동시간을 중심으로 활발하게 컴퓨터소양 및 활용 교육을 실시하고 있지만 대부분 응용소프트웨어를 활용하는 수준에 있다. 이러한 편견 현상으로 초등학생들은 소프트웨어 활용능력 습득이 컴퓨터의 전부를 이해하는 것으로 잘못 인식하여 컴퓨터 자체의 학습에 대한 흥미도를 떨어뜨리게 하는 원인이 되기도 한다.

프로그래밍 교육을 통한 학습경험은 과학 및 수학적 사고력을 향상시켜 전통적인 초등교과의 전반적 학습능력을 높일 수 있는 훌륭한 도구라 판단되며 많은 설문대상의 교사들도 공감하고 있다. 그러나 프로그래밍교육에 필수적인 국가적 교육과정은 아직 없다.

본 연구는 지금까지 학습해 왔던 컴퓨터교육인 인터넷 사용, 응용소프트웨어 습득 중심의 교육에서 벗어나 컴퓨터 자체를 이해하고 나아가 주어진 문제를 스스로 해결하고 논리력을 향상시킬 수 있는 방안으로 프로그래밍 교육과정을 개발하여 제안하였다.

제안한 본 교육과정은 인지능력이 가장 뛰어난 초등학교 5, 6학년생들이 재량활동·특별활동시간에 교수가 선택하여 활용할 수 있도록 구성하였다. 또한, 개발한 교육과정을 기초로 교재를 쉽게 만들 수 있도록 차시, 영역, 세부영역과 학습주제 그리고 학습내용을 상세히 구성하였다. 제안한 교육과정이 초등학생 5, 6학년생들에게 적합한지를 평가해 보기 위해 재량활동시간에 직접 실험학습과 자기평가 및 설문으로 다음과 같은 구체적인 결과를 얻었다.

첫째로, 프로그램 학습을 통해 초등학생들이 컴퓨터의 구조와 작동원리 이해가 가능하였다. 둘째로, 컴퓨터가 학습 매체만이 아닌 학습대상으로도 인식하게 되었다.

셋째로, 간단한 프로그래밍 학습을 통해 프로그래밍이 어렵다는 선입견을 탈피하게 되었으며 프로그래밍에 대한 관심도가 높아지게 되어 자신이 필요로 하는 소프트웨어를 만들려는 강한 의지와 동기가 발생하게 되어 전통적인 교과학습을 창의적이고 능동적인 교과 학습으로 유도하는데 도움이 되었다.

넷째로, 국가의 미래를 결정할 신기술인 IT교육이 영재교육을 선도하는 일본, 인도, 미국처럼 초등학교 컴퓨터교육에서 조기에 적용할 수 있다는 가능성을 확인했다.

이번 연구는 도시에서 선택한 일부 학생들을 대상으로 10차시분의 교육과정을 기초로 프로그래밍 학습을 실시한 결과를 분석하였다. 이것은 전체 학생들의 프로그래밍 학습에 대한 반응을 분석하는데는 다소 무리가 있다. 향후, 제안한 60차시 전체 교육과정을 기초로 농어촌 및 도시의 5, 6학년 학생들을 상대로 장기적으로 프로그래밍 교육을 실시하고 결과를 정밀하게 분석한 후 피드백시켜 제안한 교육과정을 최적수준으로 개선할 필요가 있다.

## 참고문헌

- 정재열.최재혁.공영태(2005). 한·일·인도 컴퓨터 교육과정의 비교 및 문제점 제시를 통한 우리 교육과정의 개선방안. 컴퓨터교육학회지 제1권제1호통권제1호  
National Academy of Science, <http://www.nationalacademies.org/subjectindex/com.html>  
,컴퓨터월드(2005), 컴퓨터교육의 현실과 대안, pp. 44~45
- 문외식(2002). 초등학교 ICT 활용을 위한 컴퓨터 교육과정 모델. 교육과정평가연구 제5권제1호, pp. 221~233
- 이경화.김갑수(2002). 초등학생을 위한 로고 프로그래밍 지도방안. 한국정보교육학회 2002 하계학술발표논문집 제7권2호, pp.303~310
- 허희옥외(2001). 컴퓨터교육방법 탐구. 서울:교육과학사
- 김미량(2002). 컴퓨터 프로그래밍 교육에 적용 가능한 효과적인 교수방법의 탐색적 대안. 한국컴퓨터교육학회논문지 제5권제3호, pp. 1~9
- 교육인적자원부(2003). 특별활동지도서(계발활동). 서울: 교원대학교 특별활동 국정도서 편찬위원회 p.26, p.53
- 교육인적자원부(2005). 7차교육과정자료.  
[http://www.moe.go.kr/data/policydata.htm?catmenu=m05\\_02\\_01](http://www.moe.go.kr/data/policydata.htm?catmenu=m05_02_01)
- 강성원.이애정.이재호(2003). 초등정보과학영재용 프로그래밍 교육(비주얼베이직을 이용한 접근). 한국정보교육학회 제7권3호, pp..363-371
- 문외식(1999). 쉽게 배우는 비주얼베이직 6 해설. 서울: 이한출판사.
- 김응석(1999). 한글 비주얼베이직 6. 서울: 대림출판사.
- Allen Tucker et al.(2003). A Model Curriculum for K-12 Computer Science: Final report of the ACM K-12 Task Force Curriculum Committee October, 2003. CSTA
- Computer Curricula 2004 Overview report. A Guide to Undergraduate degree programs in Computing, A cooperative project of ACM/AIS/IEEE, 22 November, 2004