

# 초등 정보영재를 위한 프로그래밍 언어 교육시스템

이영화\*, 전우천

서울수색초등학교\*, 서울교육대학교 교육대학원 컴퓨터교육학과

movielee2@hanmail.net\*, wocjun@ns.snue.ac.kr

## A Programming Language Teaching-Learning System for the Information-Gifted In Elementary Schools

Young-Hwa Lee\*, Woo-Chun Jun

Susaek Elementary School\*, Dept. of Computer Education, Seoul National University of

Education

### 요약

국가적 차원에서 추진하는 영재교육의 목적은 개인의 자아실현과 더불어 국가와 사회에서 필요로 하는 고급 인재를 양성하는 데 있다. 특히, 정보화 시대를 이끌어 갈 정보영재의 발굴과 교육은 더욱 중요하다. 컴퓨터 교육 중 프로그래밍 교육은 컴퓨터 소양, 창의적 사고와 문제 해결력, 수학적 사고력, 추론 능력을 신장시키는 정보 교육의 중요한 분야임에도 불구하고 정보영재를 위한 체계적인 교육 내용이 확립되어 있지 않은 상태이다. 이에 본 논문에서는 초등정보영재를 위한 프로그래밍 언어교육의 필요성을 살펴보고 프로그래밍 교육을 위한 교육내용을 구성하였으며 논리력 향상 및 문제 해결력 중심의 비주얼 베이직 교육 시스템을 제안하였다. 본 연구의 특징은 첫째, 학습자에게 실제 프로그램을 작성할 수 있는 기회를 제공한다. 둘째, 학습한 이론을 실제 프로그램 작성에 응용할 수 있도록 한다. 셋째, 프로그램상의 오류 수정 활동과 제시된 예제의 다양한 해결 방법을 통해 문제 해결력과 논리적 사고력을 향상시킬 수 있는 프로그래밍 학습 시스템을 제공한다.

### 1. 서론

21세기는 국경 없는 경쟁시대로 핵심자원인 지식과 정보를 얼마나 효율적으로 활용하고 창출하느냐에 따라 한 나라의 생존전략이 달려있다. 우수한 인재양성을 목적으로 하는 영재교육은 국가발전의 전략으로서도 큰 의미를 갖는다.

우리나라는 2000년 1월 영재교육진흥법을 공포하였고 2002년 3월 영재 교육법이 시행되었다. 이를 근거로 2002년 3월부터 초,중,고등학교에서 연구시범학교, 대학 부설 영재교육원, 일반학교 및 교육청을 중심으로 영재교육 대상자로 선발된 학생들에게 전국적으로 영재

교육을 실시하고 있다.

지식정보화 사회에서 우리나라는 초고속 기반 시설 확보나 컴퓨터 보급률 등 외적인 면에서는 과히 폭발적인 성장을 이뤄왔지만 정보영재의 조기 발굴 육성이나 내적 잠재력을 보유하기 위한 투자 등 정보영재 교육에 대한 연구가 미비하여 지식 정보 사회의 국가 경쟁력 향상을 위한 노력이 절실히 요구된다.

초등정보영재의 교육영역은 일반적으로 지식교육영역, 기능교육영역, 가치관교육영역의 3가지 영역으로 구분된다[1]. 이 중 지적영역의 프로그래밍 교육은 컴퓨터 소양, 창의적 사고와 문제 해결력, 수학적 사고력, 논리적 추론능력을 신장시키는 정보 교육의 중요한 분

야임에도 불구하고 정보영재를 위한 체계적인 교육 내용이 확립되어 있지 않은 상태이다.

이에 본 논문의 구성은 다음과 같다. 2장에서는 초등정보영재의 특징 및 우리나라 정보영재교육에 대해 살펴보고, 3장 프로그래밍 언어교육에서는 프로그래밍 언어교육의 필요성, 과정, 단계, 프로그래밍 언어 교육 내용, 언어 선정, 비주얼 베이직의 특징에 대해 설명하고, 4장에서는 논리력 향상 및 문제 해결력 향상을 위한 학습자의 수준에 맞게 내용을 구성한 프로그래밍 언어교육 시스템 부분에 대해 설명하며, 5장에서는 결론과 향후 과제를 제시한다.

## 2. 정보영재의 개념

정보영재를 정의하기에 앞서 먼저 영재에 대해 살펴보면 영재는 재능이 뛰어난 사람으로서 타고난 잠재력을 계발하기 위해 특별한 교육을 필요로 하는 사람이다. (영재교육진흥법 제2조 제1항)

영재교육은 영재를 대상으로 각 개인의 능력과 소질에 맞는 교육내용과 방법을 활용하여 실시된다. (영재교육진흥법 제2조 제2항)

또한, 영재교육의 목표는 ‘탁월한 잠재적 능력을 지닌 영재의 창의적 생산력, 도덕성, 자기 주도적인 학습 태도를 함양하고, 이를 통해 자아를 실현하고, 나아가 국가 사회 발전에 기여할 수 있는 인재를 육성하기 위한 것이다.’라고 정의하고 있다[2].

### 2.1 정보영재의 정의 및 특징

최근에 제시된 몇 가지 개념을 중심으로 정보영재의 정의를 정리해 보면 다음과 같다.

[3]에서는 “주어진 문제를 파악, 이해, 분석하고 정보통신 기술 활용능력을 바탕으로 새로운 정보를 수집, 가공, 재창출 할 수 있는 아동이다”라고 정의하였다. 한편, [4]에서는 “일반적인 지적능력, 컴퓨터에 대한 강한 호기심, 높은 창의력, 수학-언어적 능력, 과제 집

착력에 있어 모두 평균 이상의 특성을 소유한 자로 컴퓨터적 능력이 뛰어나거나 그 가능성이 있는 자”로 정의하였다.

이상의 특성들과 관련분야의 연구결과를 고려하여 본 논문에서는 정보영재를

첫째, 일반적 지적 능력, 컴퓨터에 대한 강한 호기심, 높은 창의력, 수학, 언어적 능력, 과제집착력의 요소에서 모두 평균 이상의 특성을 소유한 자,

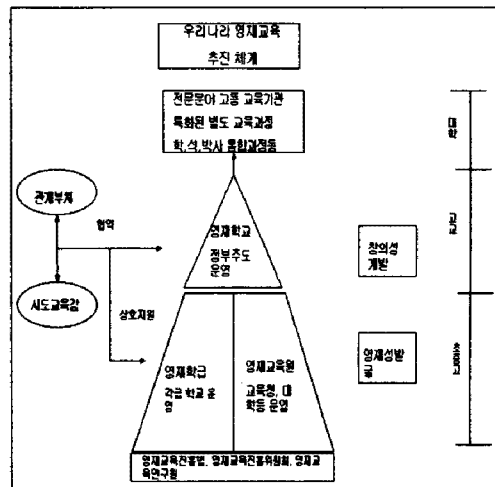
둘째, 응용 소프트웨어, 프로그래밍, 게임, 멀티미디어에 관심을 갖고 컴퓨터적 지각력, 일반화하는 능력, 추론력, 새로운 상황에 대처하는 능력, 문제를 분석하고 그들 간의 관계를 파악하는 능력이 뛰어난 자,

셋째, 컴퓨터적 표현능력, 적응력, 활용력이 뛰어나고 정보 분야에 무한한 가능성과 잠재력을 갖고 있는 자로 정의한다.

### 2.2 우리나라 정보영재교육

우리나라에서의 초등학교 영재교육은 최근 대학 부설 영재 교육원을 비롯한 극소수의 영재교육센터에서 이뤄지고 있다.

다음의 <그림 1>은 우리나라 영재교육 추진체제를 보여주고 있다.



<그림 1> 우리나라 영재교육 추진체계 [2]

또한 선진국에서는 이미 이뤄지고 있는 일반적인 초등학교 컴퓨터 교육이 우리나라에서

는 2001년에야 비로소 필수화되었다. 이에 초등학교 정보영재교육은 물론 일반적인 초등학교 컴퓨터 교육조차 체계가 잡히지 않은 단계라 할 수 있다. 초등학교 컴퓨터 교육 내용을 보면 컴퓨터 활용법 위주로만 방향이 잡혀있다. 진정한 정보영재 교육이 이뤄지려면 컴퓨터를 이해하고 나아가 활용, 응용할 수 있도록 교육되어야 하는데 상품화된 소프트웨어의 활용법만 가르치는 것은 문제점이 아닐 수 없다. 초등학생 누구에게나 컴퓨터의 기본 원리, 프로그래밍 교육 등이 체계적으로 이뤄져야 하며 특히 정보영재아들에게는 시스템 관리 교육, 고급 수준의 프로그래밍 교육 등이 이뤄져야 한다[5].

### 3. 프로그래밍 언어교육

흔히 보고서를 작성하기 위해 워드프로세서를 사용하고 어떤 문제를 풀기 위해 스프레드시트를 또 웹을 서핑하거나 음악을 다운로드 받아서 듣거나, 컴퓨터 게임을 하기 위해 소프트웨어를 사용한다.

프로그램이란 소프트웨어의 또 다른 이름으로 사람이 원하는 작업을 컴퓨터가 수행하게 하고 또 원하는 결과를 만들어 내게 하는 단계적 명령어의 집합이다. 컴퓨터 프로그래밍은 주어진 과제의 수행과 그 과정에서 당면하는 문제점의 해결을 위해 소단위로 나뉜 단계적 작업들을 점차적으로 구축해 가는 작업이라고 볼 수 있다. 즉 프로그래밍 언어는 기계가 읽을 수 있고 사람이 읽을 수 있는 형식으로 계산을 서술하기 위한 표기 체계이다[6].

Pea와 Kurland는 컴퓨터 프로그래밍에서 자주 사용되는 인지적 측면의 요구사항을 다음과 같이 제시하고 있다[7].

- 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.

- 유추적 추론 기능(Analogical Reasoning

Skills): 유추적 추론 능력이 부족하면 프로그래밍과 관련된 지식과 능력을 지니고 있으면서도 이를 프로그래밍 작업과 관련시키거나 프로그래밍을 통해 얻은 지식이나 기능을 다른 분야나 상황에 전이시키지 못한다.

- 조건적 추론 기능(Conditional Reasoning Skills): 프로그래밍에서는 명령어의 반복적 실행, 입력 정보 점검과 검토와 같은 기능을 실행 할 때 조건문을 사용하게 된다.

따라서 조건문을 이해하고 사용하는 것은 프로그래밍의 주요 부분이다.

그 외에 일시적 추론 기능, 절차적 사고(Procedural Thinking), 수학적 능력이 요구된다.

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 체계를 설계, 구축하는 입장에 놓이게 된다.

프로그래밍은 전공한 사람들만의 전유물이 아니라 모든 사람이 해야 될 기초학문이다. 수학의 수식계산, 유전공학, 생명공학, DNA 해석, 전자공학 분야의 반도체 설비, 체육 분야의 동작분석, 상대팀 전력분석, 기계공학의 기계제어, 회계분야의 회계프로그램 등은 프로그래밍의 적용 예라 할 수 있다.

#### 3.1 프로그래밍 교육의 필요성

교육적인 측면에서 프로그래밍의 의의는 다음과 같이 두 가지로 요약할 수 있다. 첫째, 프로그래밍 활동을 통해 학습자 스스로 사고력을 향상시킬 수 있다는 점과 둘째, 교과서의 문제들을 프로그래밍을 통해 해결할 수 있는 능력을 갖는 것은 정보화 사회에 대비하는 지름길이라는 점이다. 여기에서 사고력이란 합리적 사고력, 창의적 사고력, 확산적 사고력 등을 의미하는 포괄적 용어이다[8].

이러한 관점에서 컴퓨터에 관한 교육의 차원에서 현재 초등학교 학생들에게 응용 프로

그럼 사용법보다 프로그래밍이 필요한 이유는 다음과 같다[9].

첫째, 학습자에게 다양한 활동을 조장하고 문제 해결력을 신장시킨다.

둘째, 컴퓨터의 내부 동작을 이해하는데 많은 도움이 되며, 궁극적으로 컴퓨터를 보다 깊이 이해하고 활용할 수 있도록 해준다.

셋째, 각종 응용프로그램을 더욱 잘 알고 활용할 수 있는 기초를 닦아준다.

넷째, 컴퓨터 과학의 이해에 도움을 준다.

### 3.2 프로그래밍 과정

프로그래밍 작성과정을 살펴보면 5가지 주요단계가 있다.

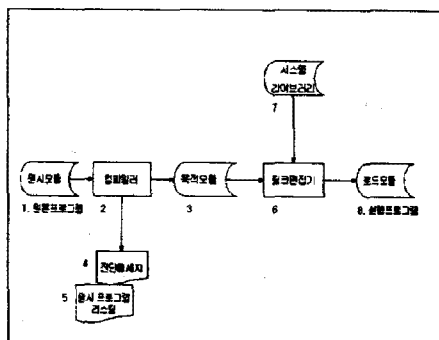
1) 문제 정의: 문제를 정의하는 것은 입력데이터가 무엇이고 그것으로부터 얻고자하는 출력, 결과가 무엇인지 결정하는 것이다.

2) 해법 설계: 문제를 푸는 상세하고 단계적인 절차인 알고리즘을 설계하는 것이다. 상용 기법은 순서도와 의사코드이다.

3) 프로그램 코딩: 의사코드나 순서도 등으로 작성된 알고리즘은 프로그래밍 언어로 옮긴다. 프로그래밍 언어에는 사람이 쓰는 언어와 마찬가지로 문법적인 규칙, 즉 구문(Syntax)이 있다.

4) 프로그램 테스트: 새로 작성한 프로그램에는 오류가 있을 수 있으므로 프로그램을 코딩한 후에는 컴퓨터에서 테스트 해보아야한다.

다음의 <그림 2>는 프로그램 테스트 과정을 보여주고 있다.



<그림 2> 프로그램 테스트 과정[10]

5) 프로그램 문서화: 프로그래밍 전 과정에 걸쳐 이뤄진다. 문서화는 프로그래밍 각 단계에서 만들어진 자료로 구성된다. 상세한 문제점의 프로그램 순서, 원시 프로그램 내부의 구성, 테스트 절차가 이 시점까지 개발된 문서 패키지의 대부분이다.

### 3.3 프로그래밍 언어의 단계

오랜 기간 동안 수백의 프로그래밍 언어들이 설계되고 구현되어 왔으나 프로그래머들은 오히려 하나 또는 두 가지 언어에만 집착하고 있다. 서로 다른 언어들을 비교, 연구함으로써 얻을 수 있는 장점은 다음과 같다[6].

첫째, 현재 사용하고 있는 언어를 더욱 잘 이해하게 된다.

둘째, 유용한 프로그래밍을 구사할 수 있는 능력을 증대시킨다.

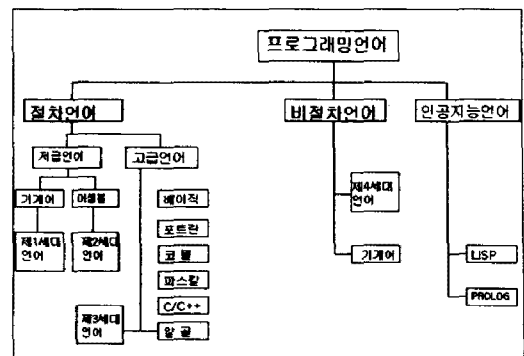
셋째, 프로그래밍 언어를 선택할 수 있는 능력을 증대시킨다.

넷째, 새로운 프로그래밍 언어를 배우기 쉽게 해 준다.

다섯째, 새로운 프로그래밍 언어를 설계하기 쉽게 해 준다. 다음은 프로그래밍의 발달단계와 특징을 알아보았다.

프로그래밍 언어는 크게 절차언어 (저급언어와 고급언어)와 비 절차 언어, 인공 지능 언어로 구분한다.

다음의 <그림 3>은 프로그래밍 언어의 단계를 보여주고 있다.



<그림 3> 프로그래밍 언어 단계[11]

### 1) 저급언어

사용자가 이해하고 사용하기에는 불편하지만 컴퓨터가 처리하기에 편리한 컴퓨터 중심의 언어이다.

기계어는 0과 1의 2진수만을 사용하여 명령어와 데이터를 나타내는 기계 중심의 언어로서, 컴퓨터가 직접 이해할 수 있어 실행 속도가 매우 빠르다.

어셈블리어는 기계어 보다 사람이 기억하기 쉬운 기호를 정하여 명령어와 데이터를 표현한다.

### 2) 고급언어

컴퓨터의 하드웨어에 대한 전문적인 지식이 없어도 배우기 쉽고, 컴퓨터 기종에 관계없이 공통적으로 사용할 수 있으며, 자연 언어에 가깝기 때문에 사람이 그 의미를 쉽게 이해할 수 있는 사용자 중심의 언어이다.

### 3) 비절차 언어

절차 언어에 비해 우리의 일상생활에서 사용하는 자연 언어에 가깝고 대화 형식으로 표현된다. 이 언어는 원하는 정보만을 표현하고 그 정보를 처리하는 방법은 표현하지 않기 때문에 사용자 입장에서 매우 편리하게 만들어졌다. 데이터베이스의 SQL언어가 대표적인 예이다.

### 4) 인공지능 언어

인간의 지능을 표현하기 위해 개발된 언어로써 추론 기능이 포함되어 있다.

LISP, PROLOG가 대표적인 인공지능 언어이다.

<표 1>은 프로그래밍 교육 내용 요소이다.

<표 1> 프로그래밍 교육 내용 요소

구분	언어 공통적인 규칙
연산자	·연산자- 단항, 산술, 쉬프트, 관계, 비트논리, 일반논리, 3항 대입 등 ·연산자 우선 순위
변수	·변수의 정의, 선언 및 사용 ·변수명, 선언, 참조(reference), 값(value)
상수	·상수 - 키코드, 수학 상수
수식	·산술식, 논리식, 문자열식
제어문	·조건문-If문, else문, else if문, switch case문 ·반복문, -while문,for문,do while문
자료형	·형선언, 열거 자료형, ·기본자료형 - 수, 논리형(Boolean), 문자 ·구조적 자료형 - 배열 ·형 변환
함수	·함수의 정의, 호출 ·사용자 정의 함수

## 3.4 프로그래밍 언어 교육내용

초등학교 현장에서 프로그래밍 교육의 실시에 앞서 다음과 같은 사항들이 고려되어야 한다.

첫째, 올바른 프로그래밍 언어의 선택이다. 고급언어가 어셈블리 언어와 기계어보다는 훨씬 배우기 쉽기 때문에 고급언어를 선택하는 것이 좋고 가급적이면 학생들의 흥미를 불러일으키기 위해 그래픽 환경을 제공하는 언어가 적합하다고 본다. 왜냐하면 프로그래밍이 기계적인 일의 반복이 아니기 때문에 자칫하면 학생들이 컴퓨터는 다루기 어려운 기계라는 인식을 갖기 쉽기 때문이다.

둘째, 이를 전담할 수 있는 교사의 양성에 있다. 프로그래밍은 응용 프로그램과는 달리 단시일에 배울 수는 없다. 따라서 이를 가르칠 수 있는 전문교사의 확보와 교육이 선행되어야 할 것이다.

프로그래밍 교육내용의 선정은 매우 까다롭고 어려운 문제이다. 초등정보영재에게 어떤 내용을 선정해서 가르칠 것인가에 대한 질문은 어떤 내용을 가르치면 가장 효과적인가에 대한 질문과 유사하다. 이를 위해 다음과 같은 방식으로 초등정보영재를 위한 프로그래밍 교육과정 내용을 추출하였다.

첫째, 프로그래밍 언어론 관련 서적에서 제

시된 내용 중 초등정보영재에게 꼭 필요한 내용이라고 판단되는 내용을 선정한다.

둘째, 각종 언어 관련 입문 서적에 설명되어있는 프로그래밍 기초 요소 중 중복되는 부분을 찾아 이를 프로그래밍 교육 내용으로 선정한다.

결국 초등정보영재를 위한 프로그래밍 교육 과정 수립을 위해서는 프로그래밍 언어에 대한 더 많은 연구가 필요할 것이다.

본 논문에서는 프로그래밍 언어론 관련 서적과 각종 프로그래밍 언어용 기초교재에서 중복되고 겹쳐지는 교육내용 요소로서 <표 1>과 같은 프로그래밍 교육내용 요소를 추출할 수 있었다.

### 3.5 프로그래밍 언어 선정

그렇다면 초등정보 영재의 프로그래밍 언어의 교육을 위해 어떤 언어를 선정할 것인가?

초등 정보영재를 위한 교육용 프로그래밍 언어 선정기준은 다음과 같다.

첫째, 학습하기가 쉬워야한다. 언어적인 속성을 기본적으로 가지고 있는 언어이면서 다른 언어에 비해 학습하기가 수월하다면 그 언어는 초등정보영재에게 적합한 교육용 언어라고 볼 수가 있다.

둘째, 비주얼한 환경을 제공해야한다. 비주얼한 환경이라는 것은 마우스를 이용해서 화면을 설계할 수 있으며 아이콘을 클릭하는 과정과 풀다운 메뉴 등의 클릭을 통해서 코딩할 수 있는 환경을 제공해야 한다는 것이다. 단순히 화면만 비주얼한 것이 아니라 학습자가 실제 함수라든가 명령어들을 외워서 직접 타이핑할 필요가 없이 풀다운 메뉴라든지 다른 방식의 마우스 접근을 통해서 프로그램을 작성할 수 있어야한다는 것이다.

셋째, 학습자에게 흥미와 동기부여가 잘 이루어질 수 있어야한다. 같은 언어라고 할지라도 학습 결과물이 보다 흥미롭고 동기 부여될 수 있는 여지와 가능성이 높다면 그 실습언어는 다른 언어보다 정보영재에게 적합하다고

볼 수 있다.

넷째, 학습자 수준과 관심에 부합되는 언어여야 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 초등정보영재에게 적합한 언어라고 볼 수 있다.

다섯째, 도움말 기능이 잘 제공되어야한다. 이것은 코드 힌트, 코드 샘플 등이 저작환경에서 제공되어야한다는 것이다[12].

위의 프로그래밍 언어 선정 기준에 따라 본 연구자는 초등정보영재의 프로그래밍 언어로 비주얼 베이직을 선정하였다.

### 3.6 비주얼 베이직의 특징

'Basic'부분은 BASIC (Beginner's All-Purpose Symbolic Instruction Code)언어를 말한다. 즉, 비주얼 베이직은 쉽다는 뜻이고 누구나 배울 수 있는 프로그램이다.

비주얼 베이직의 특징을 살펴보면 다음과 같다[13].

첫째, 비주얼 (Visual)한 프로그래밍이다. 비주얼은 눈에 보인다는 뜻으로 실행결과를 미리 보면서 프로그램을 작성한다.

둘째, 이벤트 중심 프로그래밍이다. 이벤트란 작성된 프로그램이 실행될 때 발생하는 각종 사건으로 마우스의 움직임, 클릭, 키 입력, 폼의 크기 변경 등을 이용해 자신이 원하는 처리를 하는 방식으로 프로그램을 작성한다.

셋째, 사용자가 컨트롤을 배치하고 속성을 변경하는 작업만으로 프로그램을 쉽게 작성할 수 있는 그래픽 중심의 프로그래밍 도구로 칼라 결과 뿐 아니라 디자인한대로 출력결과를 얻을 수 있어 프로그래밍을 처음 접하는 학생들에게 흥미를 잃지 않고 쉽게 배울 수 있다

넷째, 프로그램의 내부 코드는 베이직으로 작성되기 때문에 문법적으로 이해가 쉽다.

다섯째, 객체 지향 프로그램이다.

여섯째, 데이터 액세스 기능을 사용하여

Microsoft SQL Server와 기업 차원의 데이터베이스 형식에 대해 응용 프로그램을 작성할 수 있다.

일곱째, 응용 프로그래밍에서 제공되는 기능을 사용할 수 있어 게임도 쉽게 만든다.

여덟째, 인터넷 기능을 사용하면 응용 프로그램에서 인터넷을 통해 다른 문서와 응용 프로그램에 쉽게 접근할 수 있으며, 인터넷 서버 응용 프로그램도 만들 수 있다.

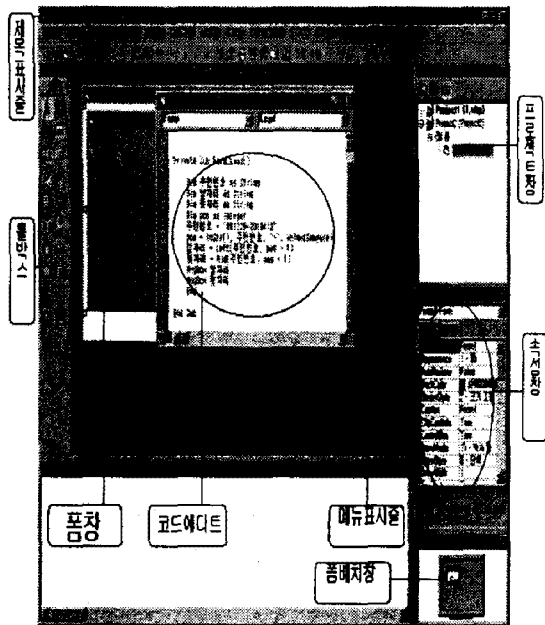
아홉째, 완성된 프로그램을 손쉽게 실행파일(exe)로 변환할 수 있다.

이러한 구성요소를 갖추었기 때문에 초등학교 학생들이 프로그램에 호기심을 가질 때 부담 없이 배울 수 있다.

접근 방법은 기초 문법을 익히고 익힌 문법을 간단한 프로그램으로 연결해 봄으로써 성취감도 느낄 수 있다.

특히 비주얼 베이직 프로그램은 폼으로 실행결과를 바로 볼 수 있기 때문에 흥미를 갖게 된다.

<그림 4>는 통합개발 환경을 보여주는 비주얼 베이직의 화면 구성을 보여주고 있다.



<그림 4> Visual Basic 화면구성

#### 4. 프로그래밍 언어 교육 시스템

본 논문에서는 처음 프로그래밍 언어를 학습하는 초등 정보영재에게 비주얼 베이직을 선정하여 프로그램의 설치, 프로그램 통합 개발 환경의 사용방법, 프로그램의 작성 방법 등의 내용을 효과적으로 학습하고 예제 학습 프로그램 중심의 모듈별 학습을 통해 학습과제를 스스로 해결하도록 설계하였다.

이 중에서 각 언어마다의 프로그래밍 공통 요소인 반복문, 입출력문, 함수, 변수, 배열요소를 비주얼 베이직 교육 시스템의 내용으로 선정하였다. 이 교육 내용으로 본 논문은 다음과 같이 설계하였다.

첫째, 학습자에게 실제 프로그램을 작성할 수 있는 기회를 제공한다.

둘째, 학습한 이론을 실제 프로그램 작성에 응용할 수 있도록 한다.

셋째, 프로그램상의 오류 수정 활동을 통해 학생 자신의 생각을 구체적인 결과물로 시각화시킬 수 있는 단계별 오류수정 학습 모듈을 설계한다.

넷째, 제시된 예제를 다른 코딩 방법으로 유도, 해결하도록 한다.

이와 같은 방법을 통해 문제 해결력과 논리적 사고력을 향상시킬 수 있는 프로그래밍 학습 시스템을 제공해 주고자 한다.

다음은 반복문, 입출력문, 함수, 변수, 배열 등 각각의 요소에 대한 정의, 문법형식, 학습 내용을 제시하였다.

##### 4.1 반복문

- 1) 정의: 특정한 명령문을 해당 조건식에 만족할 동안 반복 실행하는 것을 말한다.
- 2) 문법 형식:
  - For Next
  - Do While Loop
  - Do Until Loop
- 3) 학습내용: 프로그램 작성 후 숫자 바뀌 실행으로 반복문 원리를 안다.

다음의 <표 2>는 For Next의 프로그램을 코딩한 화면이다.

<표 2> For Next

프로그램 예제	프로그램 분석 및 처리결과
Sub main() Dim i,j,cnt As Integer cnt=0 For i=1 To 5 For j=1 To 5 cnt =cnt +1 Debug.Print Format(cnt, "@@@"); Next Debug.Print "" Next End Sub	처리 결과 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
· 학습 응용 : 숫자 5를 다른 값으로 바꿔 보기 7행의 ; 없애고 결과 확인하기 9행을 없애고 결과 확인하기	

다음의 <표 3>은 Do While Loop의 프로그램을 코딩한 화면이다.

<표 3> Do While Loop

프로그램 예제	프로그램 분석 및 처리결과
Sub main() Dim Sum, Cnt As Integer Cnt=0 Do While Cnt<10 Cnt=Cnt +1 Debug.Print "Cnt=";Cnt Loop End Sub	처리 결과 Cnt=1 Cnt=2 Cnt=3 Cnt=4 Cnt=5 Cnt=6 Cnt=7 Cnt=8 Cnt=9 Cnt=10

다음의 <표 4>는 Do Until Loop의 프로그램을 코딩한 화면이다.

<표 4> Do Until Loop

프로그램 예제	프로그램 분석 및 처리결과
Sub main() Dim Sum, Cnt As Integer Cnt=0 Do Until Cnt > 9 Cnt=Cnt +1 Debug.Print "Cnt=";Cnt Loop End Sub	처리 결과 Cnt=1 Cnt=2 Cnt=3 Cnt=4 Cnt=5 Cnt=6 Cnt=7 Cnt=8 Cnt=9 Cnt=10
· 학습 응용 : Do While ~ Loop와 Do Until 예제의 숫자 9를 10으로 바꿔보고, Do While ~ Loop와 Do Until ~의 차이점을 생각해보기	

#### 4.2 입출력문

1) 정의: 디스크와 같은 저장 매체를 이용하는 여러 가지 데이터들을 저장하고, 읽어오는 작업을 말한다.

2) 새 가지 처리 방식: 순차 파일 방식, 랜덤 파일 방식, 이진 파일 방식

3) 문법 형식: Open 경로명 For 모드 As 파일번호

- Get # 파일 번호, 레코드 번호, 변수명
- Line Input # 파일번호, 레코드 번호, 변수명
- Print # 파일번호, 출력될 문자열
- Close # 파일번호

4) 학습내용: C:\점수.txt파일의 첫째 줄에 "90", 둘째 줄에 "80", 셋째 줄에 "70"이 저장 되도록 프로그램을 작성한다.

다음의 <표 5>는 입출력 프로그램을 코딩한 화면이다.



<표 5> 입출력 코딩 화면

프로그램 예제	프로그램 분석 및 처리결과
<pre>Private Sub Form_Load() Dim fh fh = FreeFile Open "C:\점수.txt" For Output As #fh Print #fh, "90" Print #fh, "80" Print #fh, "70" Close #fh End Sub</pre>	<p>처리 결과</p> <p>"C:\점수.txt" 내용</p> <p>90</p> <p>80</p> <p>70</p>
<p>· 학습 응용 : · 순차파일에서 하나의 행을 읽어서 변수에 저장하고, 변수의 내용을 출력하기</p> <p>· 랜덤파일에서 하나의 행을 읽어서 변수에 저장하고, 변수의 내용을 출력하기 - Get # 파일번호,행번호, 필드구성</p> <p>■■■■■ Get # 파일번호,행번호, 필드구성</p>	

### 4.3 변수

1) 정의: 변수는 처리할 데이터를 저장하기 위한 컴퓨터상의 기억 공간이다.

2) 변수의 선언 : 변수의 선언은 컴파일러에게 특정 유형의 기억 공간을 확보하라는 지시어이다.

· 변수 선언 방법: Dim 변수명 As 데이터 형식

3) 학습내용: 논리식비주얼 베이직과 비주얼 C++로 표현된 논리를 비교해보고 논리를 표현하는데 일상생활에서 사용하는 글과 프로그래밍 언어 사이에는 어떤 차이가 있는지 4가지 이상 쓴다.

다음의 <표 6>은 논리식 비주얼 베이직과 비주얼 C++을 코딩한 화면이다.

<표 6> 비주얼 베이직과 C++ 코딩 화면

논리식 비주얼 베이직	논리식 C++
<pre>Sub main() Dim A As Integer Dim B As Integer Dim C As Integer Dim D As Integer Form1.Show A=InputBox("입력1") B=InputBox("입력2") C=A+B D=A*B Form1.Print C,D End Sub</pre>	<pre>#include &lt;iostream.h&gt; void main() { int A, B, C, D; cout &lt;&lt; "입력1"; cin &gt;&gt; A; cout &lt;&lt; "입력2"; cin &gt;&gt; B; C=A+B; D=A*B; cout &lt;&lt; C &lt;&lt; D;</pre>
<p>· 학습 응용 :</p> <p>· 두 프로그램은 같은 결과를 나타낸다. 두 프로그램은 무슨 논리를 표현한 것인지 글로 써 보자.</p>	

### 4.4 함수

1) 정의: 한 개 이상의 데이터를 입력받아 정해진 연산을 수행하여, 그 결과값을 되돌려 주는 명령어의 집합이다.

2) 문법 형식:

· Left (문자열, 추출 개수): 문자열의 왼쪽에서 n(추출개수)개의 문자를 취한다.

· Mid (문자열, 추출 시작 위치, 추출 개수) : 문자열의 특정 위치에서 시작해서 n(추출 개수)개의 문자를 취한다.

· Instr · Val(문자): 문자형 데이터를 숫자형 데이터로 변환한다.

· Format: 사용자 정의 숫자 형식을 만드는데 사용될 수 있는 문자들로 0, #, ,, &, / 등이 있다.

· Str(숫자): 숫자형 데이터를 문자형 데이터로 변환한다.

3) 학습내용: 주민 번호를 생년월일과 일련 번호로 나눠 출력하기.

다음의 <표 7>은 함수로 주민번호 출력을 위해 코딩한 화면이다.

<표 7> 함수 코딩 화면

프로그램 예제	프로그램 분석 및 처리 결과
<pre>Private Sub Form_Load()     Dim 주민번호 As String     Dim 앞자리 As String     Dim 뒷자리 As String     Dim pos As Integer     주민번호 = "801226-2018412"     pos = InStr(1, 주민번호, "-", vbTextCompare)     앞자리 = Left(주민번호, pos - 1)     Compar     뒷자리 = Mid(주민번호, pos + 1)     MsgBox 앞자리     MsgBox 뒷자리 End End Sub</pre>	<p>처리 결과 801226</p> <p>2018412</p>
<p>· 학습 응용 : 지정된 형식이 문자열로 변환 출력 알기 Format(1234.56, ##,##0.00")</p>	

#### 4.5 배열

1) 정의: 같은 형식의 데이터를 한 번에 여러 개 지정할 필요가 있을 때 사용하는 것이다.

2) 문법 형식: Dim 배열명(개수) As

3) 학습내용: 데이터 배열 사용으로 성적처리  
다음의 <표 8>은 배열로 성적 처리하기 위해 코딩한 화면이다.

<표 8> 배열 코딩 화면

프로그램 예제	프로그램 분석 및 처리 결과
<pre>Sub main()     Dim 총점(10) As Integer     Dim 국어(10) As Integer     Dim 영어(10) As Integer     For i=1 To 10         국어(i)=i         영어(i)=i         총점(i)=국어(i)+영어(i)     Next     For i=1 To 10         Debug.Print "총점("; i;)=":총점(i)"     Next End Sub</pre>	<p>1차원 배열선언 배열에 값을 대입 처리결과</p> <p>총점(1)=2 총점(2)=4 총점(3)=6 총점(4)=8 총점(5)=10 총점(6)=12 총점(7)=14 총점(8)=16 총점(9)=18 총점(10)=20</p>
<p>· 학습 응용 : 주어진 숫자의 데이터 배열을 사용하여 최대값과 최소값 구하기 시험 점수에서 1등의 점수 구하기</p>	

#### 5. 결론 및 향후 과제

지금까지 정보영재에 대한 일반적인 정의 및 특징을 살펴보고 우리나라 정보영재 교육을 살펴보았다. 현재까지는 정보영역이 새롭게 부각되면서도 초등학교 정보영재교육을 위한 체계적인 교육과정이 개발되어 있지 않아 정보영재 교육에 어려움이 많다.

이에 본 논문은 초등 정보영재의 컴퓨터 교육영역인 지식교육, 기능교육, 가치관교육의 세 가지 영역 중 지식교육영역의 컴퓨터 프로그래밍 언어 교육을 위해 관련 프로그래밍 논문 자료와 관련 서적을 분석하여 초등정보영재를 위한 프로그래밍 언어교육의 필요성을 살펴보았고, 프로그래밍 기초요소 중 중복되는 부분을 찾아 프로그래밍 교육내용을 선정하였으며, 논리력 향상 및 문제 해결력 중심의 프로그래밍의 기초능력 향상을 위해 비주얼 베이직 교육 시스템을 제안하였다.

본 시스템의 특징은 첫째, 학습자에게 실제 프로그램을 작성할 수 있는 기회를 제공하였다. 둘째, 학습한 이론을 실제 프로그램 작성에 응용할 수 있도록 하여 논리적 사고 형성과 문제 해결력을 기르게 하였다. 셋째, 프로그램상의 오류 수정 활동을 통해 학생 자신의 생각을 구체적인 결과물로 시각화시킬 수 있는 단계별 오류수정 학습 모듈을 설계하였다. 넷째, 제시된 예제를 다른 코딩 방법으로 유도, 해결하도록 하여 프로그래밍의 원리를 깨닫도록 하였으며, 문제 해결력과 논리적 사고력을 향상시킬 수 있도록 하였다.

향후 연구 과제로는 첫째, 정보영재의 수준을 고려한 문제 해결력 향상과 프로그래밍의 기초 능력 향상을 위한 교재 개발과 둘째, 이 교재를 이용한 웹 기반 비주얼 베이직 코스웨어 개발 등이 있다.

#### 6. 참고문헌

[1] 이영화·전우천, 초등학교 정보영재를 위한 교육과정 개발, 한국정보교육학회 2003년

하계 학술발표논문집, 제8권, 제 2호 pp. 69 - 79, 2003.

[2] 서울시 교육청, 서울 초, 중등 영재교육 연구회 워크숍 자료집, 2003.

[3] 유경미, 정보과학 영재에 대한 컴퓨터 교사들의 인식에 관한 연구, 한양대학교 교육대학원 컴퓨터교육, 2002.

[4] 오세균, 컴퓨터영재의 정의와 판별시스템, 성균관대학교 교육대학원 컴퓨터교육, 2002.

[5] 김종훈, 컴퓨터 영재 기반 다지기, 학지사, 2001.

[6] 원유현, 프로그래밍 언어론, 정익사, 2001.

[7] 강성원, 비주얼 베이직을 이용한 초등 정보과학영재용 프로그래밍 언어 교육내용 탐색, 영재교육학회, 2003.

[8] 김선희, 문제 해결력 향상을 위한 비주얼 프로그래밍 웹 코스웨어의 설계 및 구현, 한국교원 대학교 대학원, 컴퓨터 교육 전공, 2000.

[9] 임수은, 프로그래밍언어 학습을 위한 CAI 코스웨어의 설계 및 구현, 한국외국어대학교 교육대학원 전자계산교육, 2002

[10] H. L. Capron, J.A.Johnson, Computers Tools for an Informationa Age Edition7, 2002.

[11] <http://www.Oiae.com> , 정보논리영재 교육

[12] 오상진, 초등정보과학영재용 프로그래밍연구소교육, 영재교육학회, 2003 .

[13] 임성환, 자기 주도적 학습을 위한 웹 기반 Visual Basic 코스웨어, 충남대 교육대학원 컴퓨터 교육, 2002.