

# 소프트웨어 제작 분야의 성취 목표, 교수학습 방법 및 평가 방법에 관한 연구

정영식\* · 김철\*\*

전주교육대학교 컴퓨터교육과\*, 광주교육대학교 컴퓨터교육과\*\*

## 요약

소프트웨어 제작 교육은 미래의 IT 기술을 능동적으로 사용하고, 컴퓨터와 대화할 수 있는 프로그래밍 언어를 가르치는 교육이다. 이를 위해 본 연구에서는 영국, 미국, 인도, 에스토니아를 중심으로 프로그래밍 교육과 관련된 교육과정을 분석하였다. 또한, 그 결과를 토대로 우리나라의 초등학교와 중학교에서 프로그래밍 교육을 할 수 있도록 정보 과학의 한 영역으로서 소프트웨어 제작 분야의 성취 목표와 교수학습 방법, 평가 방법 등을 제안하였다. 본 연구에서는 초등학교 1학년부터 중학교 3학년까지를 4단계로 구분하고, 각 단계별로 2~3개의 세부 영역과 그에 따른 성취 목표를 제시하였다.

키워드 : 초등학교, 프로그래밍 교육, 소프트웨어 제작

## A Study on the Learning Objectives, Instructional Design, and Evaluation Methods in the Software Developing Education

Young-sik Jeong\* · Chul Kim\*\*

Dept. of Computer Education, Jeonju National University of Education\*,  
Dept. of Computer Education, Gwangju National University of Education\*\*

## ABSTRACT

Software developing education teaches students computer programming, which allows them to actively use IT and talk to computers. In this study, we analyzed computer science curriculum in the United Kingdom, the United States, India, and Estonia. In order to teach 1st - 9th grade students computer programming, we suggested the learning objectives, instructional design, and evaluation methods in software developing education focusing on Information Science. The objectives were divided into four phases, which were determined by age and grade level. Then, we determined 2-3 outcomes for each phase.

Keywords : Elementary School, Computer Programming Education, Software developing Education

---

교신저자 : 김철(광주교육대학교 컴퓨터 교육과)  
논문투고 : 2014-03-12  
논문심사 : 2014-03-12  
심사완료 : 2014-03-27

## 1. 서론

디지털 정보와 기술의 사용은 인간의 삶의 방식과 일하는 방식을 바꿔놓았으며, ICT 분야의 새로운 발전을 지속적으로 채택하는 것은 평생학습과 평생소득에 모두 필수적인 사항이 되었고, 개인은 ICT를 다. 이용하여 정보와 아이디어를 소비함과 동시에 설계하고 창출하는 능력까지 필요하게 되었다[7]. 따라서 지식과 정보의 양이 급속도로 증가함에 따라 학생들에게 지식 자체를 습득하기보다는 테크놀로지 리터시스를 통해 지식을 구하는 방법을 배우는 것이 21세기 학습자의 기초적인 능력이 되었다[5].

정부는 2000년대에 들어서면서 정보통신기술교육 운영지침을 마련하여 모든 초등학교에서 재량활동 시간에 ICT 교육을 의무적으로 하도록 하였다[9]. 그러나 2007개정 교육과정과 2009 개정 교육과정이 시행되면서 정보통신기술교육 운영지침은 사실상 유명무실하게 되어 초등학교에서의 컴퓨터교육이 거의 이루어지지 못하고 있으며, 특히 소프트웨어를 개발하는데 필요한 프로그래밍 교육은 7차 교육과정 이후에 초중등교육에서 거의 사라졌다[8].

최근 국내외의 소프트웨어 시장은 크게 팽창하고 있다. 국내 시장은 2011년에 246억 달러이었으나, 2012년에 256억 달러로 4.1%가 성장하였고, 해외 시장은 2011년 대비 4.9% 상승한 1조 1,709억 달러로 국내 소프트웨어 시장의 약 46배에 달하였다. 그러나 국내의 소프트웨어 산업 인력은 17만 명으로 여전히 부족한 편이다[11]. 이러한 문제를 해결하기 위해 박근혜 정부는 2013년 '제5차 국가정보화 기본계획'을 수립하여 초중고 단계별 컴퓨터 교육을 강화하고, 전담 교사를 배치하는 등 청소년 대상의 컴퓨터 소프트웨어 교육을 강화할 계획이다. 또한, 초중등 학생, 일반인 등 누구나 소프트웨어 프로그래밍, 컴퓨터 언어 등을 재미있게 배울 수 있는 온오프라인 교육 프로그램을 마련할 계획이다[10].

이러한 소프트웨어 교육은 국내뿐만 아니라, 영국, 인도, 에스토니아, 미국 등에서도 강조되고 있다. 영국은 기존의 ICT 교육과정을 'Computing'으로 변경하고, ICT 활용보다는 프로그래밍 교육을 중심으로 교육하고[4], 인도는 2000년대부터 초등학생들에게 스크

래치를 이용한 프로그래밍 교육을 실시하고 있다[12] 또한, 에스토니아는 2012년부터 'ProgeTiger'라는 프로그래밍 교육 사업을 통해 초등학생들에게 LOGO, KODU, Scratch와 같은 툴을 이용하여 연령에 맞는 프로그래밍 교육을 실시하고 있다[6].

이러한 해외 컴퓨터교육의 동향을 보다 세부적으로 살펴보면, ICT 활용 교육에서 소프트웨어 개발 교육으로 변하고 있음을 알 수 있으며, 프로그래밍 언어를 컴퓨터와 대화할 수 있는 하나의 언어로 인식하여, 조기에 프로그래밍 언어를 습득할 수 있도록 초등학생부터 프로그래밍 교육을 실시하고 있다.

그러나 우리나라의 초중등학교에서는 프로그래밍 교육이 거의 이루어지지 못하고 있으므로, '제5차 국가정보화 기본 계획[10]'에 따라 초등학교부터 프로그래밍 교육을 실시하기 위해서는 소프트웨어 제작과 관련된 교육과정이 시급하게 만들어져야 한다. 따라서 본 연구에서는 초등학교와 중학교에서 '소프트웨어 제작'과 관련된 교육과정을 개발·운영할 수 있도록 프로그래밍 교육과 관련된 해외 사례를 분석하여 소프트웨어 제작 분야의 성취 목표와 교수학습 방법, 평가 방법을 제시하였다.

## 2. 외국 사례 연구

### 2.1 영국

영국은 정보통신기술 발전에 따라 수동적인 기기 사용자가 아닌 능동적인 사용자를 양성하기 위해 정보 원리와 계산 원리에 대한 지식과 이해를 바탕으로 프로그램과 시스템, 콘텐츠를 제작하기 위한 프로그래밍 교육을 강조하고 있으며, 이를 위해 '컴퓨팅(computing)' 교과를 운영하고 있다. 컴퓨팅 과목은 크게 컴퓨터 원리(CS; computer science), 정보 기술(IT; information technology), 디지털 리터러시(DL; digital literacy)로 구분된다[2].

영국의 컴퓨팅 교과에 포함된 내용 중에서 소프트웨어 제작과 관련된 내용을 중심으로 살펴보면 다음과 같다.

첫째, 컴퓨팅 교과의 목표는 학생들이 논리와 알고리즘을 포함한 컴퓨터 과학의 기본적인 원리를 이해하고

적용할 수 있도록 하고, 문제 해결을 위해 실제적인 컴퓨터 프로그램 작성을 반복할 수 있도록 제시하고 있다.

둘째, 컴퓨팅 교과에 제시된 key stage별 내용 중에서 초등학교와 중학교에 해당하는 것을 핵심 키워드 중심으로 정리하면 <Table 1>에 제시된 바와 같다[7].

영국은 <Table 1>에 제시된 바와 같이 국가 교육 과정에 제시하여 초등학교 1학년부터 프로그래밍 교육을 실시하도록 하고 있다. 알고리즘이 무엇인지 이해하고, 알고리즘이 디지털 장치에서 프로그램으로서 어떻게 구현되는지, 프로그램은 일련의 명령어를 실행한 것임을 이해하도록 하고 있으며, 간단한 프로그램을 직접 작성하고 시험해 보도록 하고 있다. 이러한 초기 프로그래밍 교육은 수동적인 ICT 활용 교육의 문제점에서 출발하였다.

<Table 1> Computing curriculum in British

구분	학년	교육 내용
KS1	1-2	· 알고리즘과 프로그램의 이해 · 간단한 프로그램의 작성과 시험
KS2	3-6	· 목적에 맞는 프로그램 설계와 작성 · 문제를 작게 쪼개어 해결(디버깅) · 순차, 선택, 반복, 변수, 입출력 활용 · 알고리즘과 프로그램의 에러 검출과 수정
KS3	7-9	· 실세계 문제의 상태와 행위를 모델링 · 정렬과 검색, 프로서저 활용 · 2가지 이상의 프로그래밍 언어 사용 · 부울(AND, OR, NOT) 논리 이해

## 2.2 미국

미국은 2003년 ACM(Association for Computing Machinery)은 K-12 학생들을 위한 컴퓨터과학 교육 과정을 개발하였고, 2006년에는 제2판이 배포되었다. 수준을 크게 4가지로 구분하였으며, 초등학교와 중학교에 해당하는 제1수준(K-8)과 제2수준(9-10)은 컴퓨터 과학의 기초와 현대 생활 속의 컴퓨터 과학을 다루고 있다[3]. 또한, 컴퓨터 과학은 소프트웨어 응용과 컴퓨터, 컴퓨터 과학을 통한 문제해결, 컴퓨팅의 사회적 맥락 등 크게 3개 영역으로 구분되어 있다.

이 중에서 소프트웨어 제작과 관련된 영역은 ‘컴퓨터과학을 통한 문제해결’ 영역이며, 해당된 세부 주제는 ‘문제해결과 알고리즘’, ‘컴퓨터 프로그래밍’ 등 2개의 주제가 해당된다. 이들 주제를 중심으로 중점

요소를 정리하면 <Table 2>에 제시된 바와 같다.

<Table 2> Computer Science curriculum in USA

구분	문제해결과 알고리즘	컴퓨터 프로그래밍
K-2	· 알고리즘의 정의 · 간단한 작업 과정 쓰기	· 컴퓨터는 프로그램에 의해 작동됨을 이해 · 컴퓨터 프로그램, 프로그래머, 프로그래밍 언어 개념
3-5	· 성공적인 작업을 위한 알고리즘 작성과 실행 · 논리연산자 AND, OR 결과	· 프로그래밍언어의 목적 · 간단한 순차 프로그램 작성 및 시험
6-8	· 컴퓨터의 기능은 명령을 실행한 것임을 이해 · 주어진 문제를 해결하기 위해 알고리즘 작성 · 효과적인 알고리즘 선택	· 알고리즘을 프로그램으로 작성 · 주어진 문제를 해결하기 위해 변수와 반복문 이용하여 프로그램 작성과 시험
9-10	문제해결 절차 이해 표준 설계도구 활용 주어진 알고리즘이 성공적인 해결 여부 판단	· 특정 사양에 맞는 프로그램의 작성과 시험, 실행 · 문제를 프로그램으로 작성 · 적절한 자료형 선택 · 구조화된 프로그램 작성

미국의 경우에는 주어진 문제를 해결하기 위해 알고리즘으로 표현하고, 그것을 다시 프로그램으로 작성하여 실행하고 시험할 수 있도록 하고 있다. 특히 컴퓨터 프로그램의 계획과 작성, 시험 등 일련의 절차를 8학년 까지 배우도록 하고 있으며, 9-10학년에서는 여러 가지 자료형뿐만 아니라 배열 변수와 간단한 자료형을 직접 만들어 사용할 수 있도록 지도하고 있다. 또한, 서브 프로그램을 작성하여 구조화된 프로그램을 작성할 수 있도록 하며, 다양한 고급 언어를 서로 비교하고, 객체 지향 언어를 사용할 수 있도록 지도하고 있다.

## 2.3 인도

인도는 이미 2000년대부터 컴퓨터 과학을 교과서로 활용하고 있으며, 2005년에 만들어진 NCERT(National Council of Educational Research and Training) 프레임워크에 따라 1학년부터 8학년까지는 핵심 과목으로서 컴퓨터 과학을 다루고 있고, 9학년에서 10학년은 선택 과목으로 다루고 있다[12].

본 연구에서는 필수 과목으로 활용되고 있는 1학년과 8학년까지의 내용 중에서 소프트웨어 제작과 관련된 내용만을 정리하면 <Table 3>에 제시된 바와 같다. 참고로 1학년과 2학년에서는 소프트웨어 제작과 직접적으로 관련된 내용이 없으므로 본 연구에서는 제외하였다.

<Table 3> Computer Science curriculum in India

구분	What	How
3	<ul style="list-style-type: none"> <li>· 프로그램과 작업 절차</li> <li>· 스크래치로 간단한 명령과 편집</li> <li>· 컴퓨터 언어에 대한 개념</li> </ul>	<ul style="list-style-type: none"> <li>· 단계적 사고하기</li> <li>· 스크래치 소개하기</li> <li>· 스크래치로 간단한 애니메이션 만들기</li> </ul>
4	<ul style="list-style-type: none"> <li>· 제어문 활용</li> <li>· 스크래치의 스크립트 작성</li> <li>· 스크래치 블록의 인수 변경</li> </ul>	<ul style="list-style-type: none"> <li>· 스크래치 즐기기</li> <li>· 다른 주제로 애니메이션, 게임 스크립트 작성하기</li> </ul>
5	<ul style="list-style-type: none"> <li>· 간단한 프로그램 작성</li> <li>· 프로그램 다운로드와 설치</li> </ul>	<ul style="list-style-type: none"> <li>· 스크래치로 조직하기</li> <li>· 데이터 정렬과 비교하기</li> </ul>
6	<ul style="list-style-type: none"> <li>· 순차적 실행</li> <li>· 구문과 구조 논리</li> </ul>	<ul style="list-style-type: none"> <li>· 비프로그래밍 흐름도 그리기</li> <li>· HTML 기본 활용하기</li> </ul>
7	<ul style="list-style-type: none"> <li>· 간단한 알고리즘</li> <li>· 검색과 정렬</li> </ul>	<ul style="list-style-type: none"> <li>· 간단한 알고리즘 이해하기</li> <li>· 부울 논리 사용하기</li> <li>· 여러 가지 구조를 활용한 프로그램 작성하기</li> </ul>
8	<ul style="list-style-type: none"> <li>· 간단한 구조 프로그램</li> <li>· GUI보다는 구문 기반의 구조적 언어 활용</li> </ul>	<ul style="list-style-type: none"> <li>· 소프트웨어의 구조 이해하기</li> <li>· 언어의 모든 기능 연습하기</li> </ul>

인도의 경우에는 2000년부터 초등학생을 대상으로 프로그래밍 교육을 시작하였으며, 특히 초등학교 3학년년부터 스크래치를 활용하여 간단한 게임 프로그램을 작성할 수 있도록 지도하였다. 6학년에서는 HTML을 이용하여 웹 프로그램을 작성할 수 있도록 하였으며, 7학년에서는 간단한 검색과 정렬 알고리즘을 지도하였다. 8학년에서는 GUI 방식의 프로그래밍 툴보다는 텍스트 기반의 구조적 언어를 소개함으로써 향후 다른 고급 언어로의 전이를 돕고 있었다. 또한, 인형극, 역할극, 스토리텔링, 게임, 탐구학습, 시나리오 기반 학습, 연역적/귀납적 접근법, 문제해결 학습 등 다양한 교수학습전략을 학년에 맞게 사용하도록 하고 있다.

### 2.4 기타

앞에서 소개된 국가의 경우 대부분 프로그래밍 교육이 정규 교육과정에 포함되어 운영되고 있으나, 에스토니아는 정규 교육과정에 포함되어 있지 않았으나, 발달된 소프트웨어 산업을 유지하기 위하여 민간 기업이나 공공 기관을 중심으로 클럽 활동과 방과후 학교 운영으로 활발하게 프로그래밍 교육이 이루어지고 있다. 에스토니아는 프로그래밍 교육을 정규 교육과정에서는 시행하고 있지 않지만, 'Proge Tiger'라는 사업을 통해 초중등 학생들이 수업 시간이나 클럽 활동을 통해 프로그래밍을 배우고 웹 애플리케이션과

웹 사이트를 제작할 수 있도록 하고 있다[6]. Proge Tiger의 목표는 학생들의 논리적 사고력, 창의력, 수학적 능력을 개발하고, 프로그래밍이 재미있고 누구나 할 수 있다는 것을 보여주며, 실제적인 활동을 통해 기본적인 프로그래밍을 가르친다. 또한, 학생들의 연령에 맞는 프로그래밍 언어를 가르치는 것을 목표로 삼고 있다. 초등학생들에게는 LOGO, KODU, Scratch와 같은 그래픽 프로그래밍 언어만 사용되며, 중학생들은 LEGO Mindstorms과 같은 로봇을 이용한 프로그래밍 언어와, 웹디자인뿐만 아니라 웹 사이트와 웹 애플리케이션을 만드는 방법을 배운다.

이스라엘은 중학교에서부터 프로그래밍 교육이 본격적으로 이루어지고 있다. 중학교에서는 스크래치, HTML5, 로보틱스를 활용한 프로그래밍 교육이 이루어지고, 고등학교에서는 선택과목으로 '컴퓨터 과학'이 운영되고 있으며, 컴퓨터 과학을 수강한 학생을 대상으로 '소프트웨어 엔지니어링' 과정이 운영되고 있다. 컴퓨터 과학 과목에서는 2년 동안 450시간의 기본 과정을 통해 컴퓨터의 원리를 배우고, Java와 C#과 같은 고급 프로그래밍 언어를 배운다. 이후 심화과정으로 소프트웨어 엔지니어링 과정에서는 450시간 동안 실제적인 응용 프로그램을 개발하기 위해 프로젝트 기반의 수업이 이루어진다. 기본 과정을 이수한 학생 대부분은 심화과정을 이수하며, 이러한 학생은 전체 고등학생 중에서 15%에 이른다[1].

### 3. 성취 기준

초등학교와 중학교를 대상으로 한 영국, 미국, 인도 등에서 실시하고 있는 컴퓨터 교육 과정 중에서 소프트웨어 제작과 관련된 내용을 중심으로 분석한 결과[1][2][3][4][6][12], 초등학교 1~2학년에서는 주로 문제해결의 절차와 알고리즘 및 프로그램의 개념이 소개되었고, 3~4학년은 스크래치와 같은 그래픽 기반의 프로그래밍 툴을 이용하여 간단한 프로그램을 작성하도록 하였다. 또한, 5~6학년에서는 순차문, 선택문, 반복문 등과 같은 여러 가지 구문을 이용한 프로그래밍 교육을 실시하였으며, 중학교 수준에서는 다양한 자료형과 절차문을 가르치고, 정렬이나 검색 등의 알고리즘을 가르치도록 하였다. 이러한 사례를

감안하여 본 연구에서는 <Table 4>와 같이 소프트웨어 제작에 대한 내용을 4단계로 구분하고, 세부 영역을 11개 영역으로 구분하였다.

<Table 4> Core contents of software developing

단계	학교급	학년	세부 영역
1	초등학교	1-2	· 생활과 알고리즘 · 프로그램의 세계
2		3-4	· 알고리즘의 표현 · 프로그래밍의 이해 · 간단한 프로그램 제작
3		5-6	· 간단한 데이터 구조 · 프로그래밍 개발 환경 · 다양한 프로그램 제작
4	중학교	1-3	· 다양한 알고리즘 · 데이터 구조 생성 · 소프트웨어 제작 관리

### 3.1 초등학교 1-2학년(1단계)

1단계는 초등학교 1~2학년을 대상으로 적용되고, ‘생활과 알고리즘’, ‘프로그램의 세계’로 구분되며, 세부 성취 기준은 <Table 5>와 같다.

<Table 5> Achievement standards(phase 1)

영역	성취 기준
생활과 알고리즘	· 일상생활에서 일어나는 일이 순서가 있음을 이해할 수 있다. · 간단한 문제 상황을 말로 설명할 수 있다. · 알고리즘의 뜻을 이해할 수 있다. · 일이 일어난 순서에 따라 그림이나 글자를 나열할 수 있다. · 간단한 문제를 해결하기 위한 방안을 순서대로 말할 수 있다. · 순서에 따라 간단한 문제를 해결하려는 태도를 갖는다.
프로그램의 세계	· 프로그래밍된 장난감은 정확한 명령에 따라 실행됨을 이해할 수 있다. · 컴퓨터는 프로그램에 의해 작동됨을 이해할 수 있다. · 컴퓨터 프로그램은 명령들의 집합임을 이해할 수 있다. · 일상생활 속에서 사용하는 응용 프로그램의 이름을 말할 수 있다. · 프로그래밍된 장난감을 이용하여 간단한 문제를 해결할 수 있다. · 프로그래밍된 장난감을 고장 나지 않게 바르게 사용할 수 있다.

### 3.2 초등학교 3-4학년(2단계)

2단계는 초등학교 3~4학년을 대상으로 적용되고, ‘알고리즘의 표현’, ‘프로그래밍의 이해’, ‘간단한 프로그램 제작’으로 구분되며, 세부 성취 기준은 <Table 6>와 같다.

<Table 6> Achievement standards(phase 2)

영역	성취 기준
알고리즘의 표현	· 주어진 문제를 해결하기 위한 과정을 단계별로 분명하게 제시할 수 있다. · 주어진 알고리즘을 보고 다음 동작을 예측할 수 있다. · 주어진 알고리즘을 순서도로 표현할 수 있다. · 조건에 따라 문제 해결 과정이 달라질 수 있음을 이해할 수 있다. · 하나의 문제를 해결하기 위해서 다양한 알고리즘을 사용할 수 있다. · 여러 가지 알고리즘 중에서 가장 효율적인 것을 찾을 수 있다. · 문제를 해결할 때 가장 효율적인 방법을 찾으려는 태도를 갖는다.
프로그래밍의 이해	· 프로그래밍은 알고리즘을 프로그램으로 옮기는 과정임을 이해할 수 있다. · 인간의 언어처럼 여러 가지 프로그래밍 언어가 있음을 인지할 수 있다. · 프로그래밍 언어가 정확하지 않으면 에러가 발생할 수 있음을 알 수 있다. · 주어진 컴퓨터 프로그램을 알고리즘으로 표현할 수 있다. · 간단한 컴퓨터 프로그램을 읽고 그 결과를 예측할 수 있다. · 문제 해결 과정에서 오류를 찾고 해결하려는 자세를 갖는다.
간단한 프로그램 제작	· 간단한 알고리즘을 컴퓨터 프로그램으로 작성할 수 있다. · 변수문과 조건문을 이용하여 간단한 프로그램을 작성할 수 있다. · 컴퓨터 프로그램에 포함된 간단한 오류를 찾아 수정할 수 있다. · 프로그래밍 도구의 필요성과 기능을 이해한다. · 프로그래밍을 위해 간단한 프로그래밍 도구를 활용할 수 있다. · 컴퓨터 프로그램의 오류를 찾고 해결하려는 자세를 갖는다.

### 3.3 초등학교 5-6학년(3단계)

3단계는 초등학교 5~6학년을 대상으로 적용되고, ‘간단한 데이터 구조’, ‘프로그래밍 개발 환경’, ‘다양한 프로그램 제작’으로 구분되며, 세부 성취 기준은 <Table 7>과 같다.

<Table 7> Achievement standards(phase 3)

영역	성취 기준
간단한 데이터 구조	· 데이터 구조의 개념을 이해할 수 있다. · 스택과 큐의 차이를 이해할 수 있다. · 스택과 큐가 일상생활에서 사용되는 예를 설명할 수 있다. · 2차원 배열을 활용한 간단한 프로그래밍을 작성할 수 있다. · 2차원 배열을 이용하여 스택과 큐를 프로그래밍할 수 있다.
프로그래밍 개발 환경	· 콘솔 기반의 프로그램과 윈도우 기반의 프로그램의 차이를 이해할 수 있다. · 콘솔 기반의 프로그래밍 도구를 활용하여 프로그램을 작성할 수 있다. · 윈도우 기반의 프로그래밍 도구를 활용하여 프로그램을 작성할 수 있다. · 컴포넌트 방식의 프로그래밍 도구를 활용하여 프로그램을 작성할 수 있다. · 프로그래밍 도구에서 제공하는 다양한 라이브러리를 활용할 수 있다. · 프로그래밍 목적에 맞는 개발 도구를 찾아 활용하려는 자세를 갖는다.
다양한 프로그램 제작	· 프로그래밍 언어의 다양한 식문과 그 쓰임새를 이해할 수 있다. · 반복문, 선택문, 함수문을 이용하여 프로그램을 작성할 수 있다. · 논리연산자와 관계연산자를 이용하여 프로그램을 작성할 수 있다. · 프로그램의 일부분만 쪼개어 해석하고 실행할 수 있다. · 프로그램에 포함된 다양한 형태의 오류를 찾아 수정할 수 있다. · 문제해결을 위해 일부분을 쪼개어 자세히 분석하려는 자세를 갖는다.

### 3.4 중학교 1~3학년(4단계)

4단계는 중학교 1~3학년을 대상으로 적용되고, ‘다양한 알고리즘’, ‘데이터 구조 생성’, ‘소프트웨어 제작 관리’로 구분되며, 세부 성취 기준은 <Table 8>과 같다.

<Table 8> Achievement standards(phase 4)

영역	성취 기준
다양한 알고리즘	<ul style="list-style-type: none"> <li>· 다양한 데이터 정렬 방법을 이해하고, 그 차이를 설명할 수 있다.</li> <li>· 다양한 데이터 검색 방법을 이해하고, 그 차이를 설명할 수 있다.</li> <li>· 데이터 정렬 방법을 순서대로 표현하고, 최적의 방법을 찾을 수 있다.</li> <li>· 데이터 검색 방법을 순서대로 표현하고, 최적의 방법을 찾을 수 있다.</li> <li>· 순서도를 보고 오류를 찾거나 다음 동작을 예측할 수 있다.</li> <li>· 효율적인 방법으로 자료를 정렬과 검색하려는 자세를 갖는다.</li> </ul>
데이터 구조 생성	<ul style="list-style-type: none"> <li>· 다양한 형태의 데이터 구조와 자료형을 이해할 수 있다.</li> <li>· 다차원 배열을 이용한 프로그램을 작성할 수 있다.</li> <li>· 간단한 데이터 구조를 설계하고, 그것을 프로그램으로 작성할 수 있다.</li> <li>· 간단한 데이터 구조를 이용하여 특정 값을 입출력할 수 있다.</li> <li>· 파일 형태로 입출력하는 프로그램을 작성할 수 있다.</li> <li>· 필요한 자료를 추상화하고 구조화하려는 자세를 갖는다.</li> </ul>
소프트웨어 제작 관리	<ul style="list-style-type: none"> <li>· 프로그램 속의 에러를 수정하기 위해 논리적인 추론을 사용할 수 있다.</li> <li>· 소프트웨어의 개발 공정을 이해하고, 단계별로 필요한 문서를 작성할 수 있다.</li> <li>· 사용자 입력 값에 따라 결과가 달라지는 프로그램을 작성할 수 있다.</li> <li>· 디버깅 툴을 이용하여 프로그램을 수정할 수 있다.</li> <li>· 기계어의 의미를 알고, 간단한 프로그램을 작성할 수 있다.</li> <li>· 자신이 만든 프로그램을 비판적으로 개선하려는 자세를 갖는다.</li> </ul>

### 4. 교수 학습 방법

소프트웨어 제작 영역은 컴퓨터 프로그래밍 언어를 이용하여 문제 해결력을 신장시키기 위한 것으로서 주어진 문제를 파악하고, 그것을 해결하기 위한 과정을 알고리즘을 나타냄으로써 자신이 생각한 문제 해결 방법을 다른 학생들과 공유할 수 있게 지도한다. 또한, 알고리즘을 표현된 것을 컴퓨터 프로그램으로 작성하여 실제 실행하고, 그 결과를 시험하면서 자신이 생각한대로 결과가 나오는지 확인할 수 있으며, 원하는 결과가 나타나지 않았을 경우에는 그 원인을 찾아 해결하는 과정을 배우게 한다.

소프트웨어 제작에 필요한 교수학습방법은 학교의 실정과 학생의 흥미와 관심, 능력 수준을 고려하여 학습 집단을 다양하게 편성하여 지도한다. 또한, 인형극, 역할극, 스토리텔링, 게임, 탐구학습, 삽화 지도, 실제 체험, 시나리오 기반 학습, 연역적/귀납적 접근법, 문

제 해결학습, 소그룹 활동, 브레인스토밍, 상호 협력 등 학생들의 발달 능력과 학습 주제에 따라 다양한 교수 학습 방법이 활용될 수 있도록 한다. 아울러 컴퓨터 프로그래밍 교육에 필요한 툴을 선택할 때에는 발달 단계를 고려하되, 저학년의 경우에는 KODU와 게임 기반의 프로그래밍 툴을 선택하고, 고학년의 경우에는 LEGO 마인드스톰과 같은 컴포넌트 기반의 프로그래밍 툴을 사용한다. 중학생의 경우에는 다른 언어로의 전이를 위해 HTML이나 Javascript와 같은 스크립트 기반의 프로그래밍 툴을 사용한다.

각 단계별 세부적인 교수학습방법을 정리하면 다음과 같다.

첫째, 1단계에서는 일상생활에서 발생하는 간단한 문제 상황을 말로 표현하고, 그것을 해결하는 방법을 시간적인 순서로 말하게 함으로써 알고리즘의 의미를 이해하도록 지도한다. 또한, 프로그래밍된 장난감을 체험함으로써 프로그램의 존재와 의미를 이해하고, 우리가 생활하는 과정에서 발생될 수 있는 간단한 문제를 정해진 순서에 따라 해결하려는 태도를 가질 수 있도록 지도한다. 아울러, 학생들이 간단한 문제를 해결하기 위해 직접 프로그래밍을 하기보다는 문제 해결 과정을 알고리즘으로 표현하는 데에 중점을 두어 지도한다.

둘째, 2단계에서는 주어진 문제를 해결하는 데 필요한 알고리즘을 순서대로 표현할 수 있고, 순서도를 보고 가장 효율적인 알고리즘을 찾을 수 있도록 지도한다. 또한, 프로그래밍의 개념과 프로그래밍 언어의 다양성을 이해하고, 간단한 알고리즘을 프로그램으로 작성한 후에 실행할 수 있도록 하며, 실행 결과를 보면서 간단한 오류를 찾을 수 있도록 지도한다. 아울러, 학생들이 일상생활에서 발생할 수 있는 문제의 해결 방안을 순서대로 나타내고, 그것을 해결하기 위해 프로그램으로 작성하려는 태도를 갖도록 지도한다.

셋째, 3단계는 배열과 스택, 큐와 같은 간단한 데이터 구조를 이해하고, 그것을 이용한 프로그램을 작성할 수 있도록 지도한다. 프로그램 개발 도구에서 제공하는 라이브러리를 이용하여 텍스트 기반 프로그래밍과 그래픽 기반 프로그래밍을 할 수 있도록 지도하며, 반복문과 함수문 등을 이용한 프로그램을 작성하고, 오류를 수정할 수 있도록 지도한다. 아울러, 학생들이 일상생활에서 많이 활용될 수 있는 데이터 구조를 이용하여 복잡한 문

제를 보다 쉽게 해결하려는 태도를 갖도록 지도한다.

넷째, 4단계에서는 정렬, 검색과 같은 다양한 알고리즘을 이해하고, 다차원 배열과 간단한 데이터 구조를 파일 형태로 입출력할 수 있도록 지도한다. 또한, 소프트웨어 개발 공정을 이해하고, 각 단계별로 필요한 문서를 작성할 수 있으며, 전문적인 디버깅을 툴이나 간단한 기계어를 활용할 수 있게 되어 소프트웨어 개발 능력이 향상될 수 있도록 지도한다. 아울러, 학생들은 일상생활에서 자주 사용되는 정렬 방식과 검색 방법을 이해하고, 그것을 프로그래밍 함으로써 효율적인 데이터 관리를 할 수 있도록 지도한다.

## 5. 평가 방법

소프트웨어 제작 영역에서는 특정 영역 또는 특정 성취 기준에 치우치지 않도록 전반적으로 평가하되, 다음 사항에 중점을 둔다.

첫째, 기본적인 개념이나 원리, 사실 등 관련 지식의 이해 정도 응용력·창의력을 발휘하는 문제 해결 능력, 실습 능력, 도구를 바르게 사용하는 능력과 성실하게 학습에 임하는 태도 등을 평가한다.

둘째, 각 세부 영역별 성취 기준을 고려하여 학습 과정이나 결과를 수시로 평가하고, 학습 활동의 관찰, 면담 등 여러 가지 방법을 활용하되, 사전에 평가 기준, 방법, 시기 등을 계획하여 평가한다.

셋째, 평가는 문제해결 절차에 대한 말하기, 알고리즘 그리기, 프로그램 작성하기, 실행 결과 예측하기, 오류를 찾고 수정하기 등 다양한 형태로 실시한다.

넷째, 특정 프로그래밍 툴을 도입할 경우, 툴의 사용법 자체를 평가하기보다는 프로그램의 완성도를 평가한다. 또한, 프로그램의 수행 결과만을 보고 평가하기보다는 결과가 예상과 다르더라도 그 원인을 찾아 수정할 수 있는 능력이 있는지를 평가하도록 한다.

다섯째, 문제 해결에 대한 태도, 가치관의 평가는 자율적인 학습 경험을 발전시켜 나갈 수 있는 자기평가와 실천에 주안점을 두고 평가하도록 한다.

여섯째, 평가 결과는 학습 목표, 학습 지도 방법, 지도 계획 등에 반영하여 소프트웨어 제작 교육의 개선 자료로 활용한다.

## 6. 결론

해외의 컴퓨터교육과 관련된 교육과정의 변화를 살펴보면, 컴퓨터활용교육에서 컴퓨터과학교육으로, ICT 활용교육에서 ICT 소양교육을 넘어 소프트웨어 개발 교육으로, 컴퓨터 기능 습득에서 전산적 사고력(Computational Thinking) 신장으로 변하고 있다. 또한, 컴퓨터교육 관련 교과 또한 선택 교과에서 필수 교과로, 보조 교과에서 도구 교과로, 고등직업교육에서 초중등교육으로 바뀌고 있으며, 몇몇 국가에서는 프로그래밍 언어 교육을 초등학교 때부터 체계적으로 가르치고 있다.

프로그래밍 언어는 제3외국어이다. 모국어로서 한국어가 있고, 국제적으로 통용되는 제1외국어로서 영어가 있다. 또한, 특정 지역에서의 활동을 위해 아랍어나 일본어, 중국어, 스페인어 등을 제2외국어로 배울 수 있다. 그러나 프로그래밍 언어는 인간과의 대화가 아닌 컴퓨터와의 대화 언어이다. 즉, 미래의 기술과의 대화이며, 그것이 곧 프로그래밍 언어이다. 프로그래밍 언어는 인간이 사용하는 자연어와 같이 조기에 교육할수록 그 효과는 크고, 자주 활용되어야 한다. 이를 위해서는 프로그래밍 교육이 초등학교부터 시작되어야 한다[12].

그동안 초중등교육에서는 만들어진 소프트웨어를 활용하는 방법만을 가르쳐왔다. 따라서 미래를 이끌어갈 창의적인 인재를 양성하기 위해서는 소프트웨어 활용보다는 제작하는 능력을 갖게 해야 한다. 만들어진 소프트웨어를 수동적으로 사용하기보다는 문제점을 보완하고, 자신에 맞는 소프트웨어로 개발할 수 있는 능동적인 개발자로서의 능력을 갖추도록 해야 한다.

모든 아이들이 프로그래밍 교육을 받을 필요는 없다. 그러나 최소한 모든 아이들이 프로그래밍을 할 수 있는 기회를 제공해야 한다. 지식정보화시대에서 컴퓨터는 정보를 수집하고, 편집하고, 분석하고, 평가할 수 있는, 가장 효과적인 도구이며, 컴퓨터를 가장 잘 이해할 수 있는 방법은 프로그래밍이기 때문이다.

지금까지 초등학교와 중학교에서 프로그래밍 교육을 도입하기 위해 소프트웨어 제작과 관련된 내용과 교수학습방법, 평가 방법을 제시하였다. 향후 이것을 기반으로 실제 학교에서 활용되기 위해서는 다음과 같은 노력이 필요하다.

첫째, 어린 학생들이 프로그래밍을 배우는 것은 어렵

다는 생각을 버려야 한다. 에스토니아의 IT 기업인 Reaktor사는 만4세부터 9세의 어린이들을 위해 Code school을 운영하면서 텍스트 기반의 프로그래밍 언어인 Turtle Roy를 2013년부터 가르쳐 왔다. 그 결과, 교육을 받은 어린이와 그의 부모의 호응이 높아 현재는 수 십 개의 회사와 함께 Code School을 운영하고 있으며 현재는 수 백 명의 어린이들이 참여하고 있다[13]. 프로그래밍 언어 자체가 어렵다기보다는 가르치는 교재와 방법을 쉽게 해야 한다. KODU, LOGO, Scratch, LEGO Mindstorms 등 다양한 프로그래밍 툴이 있다. 이러한 툴을 연령에 맞게 선택하여 가르친다면 어린 아이들도 프로그래밍을 보다 쉽고 재미있게 배울 수 있다.

둘째, 제안된 소프트웨어 제작 영역이 실제로 학교에서 배울 수 있도록 국가 수준의 교육과정에 반영되어야 한다. 소프트웨어 교육은 이미 제5차 국가정보화 기본 계획에서 초등학교에서부터 실시하겠다고 밝힌 바 있다. 그러나 교육과정에 어떻게 반영하겠다는 계획은 구체적으로 제시된 바 없으므로 학교 현장과 전문가들의 의견을 종합하여 2015년 교육과정에 반영될 수 있도록 실질적인 조치가 이루어져야 한다.

셋째, 소프트웨어 제작 교육을 위해서는 교원 양성 단계에서부터 관련 교육이 이루어져야 한다. 이를 위해서는 교육대학교의 교육과정에 소프트웨어 제작 교육에 필요한 강좌가 개설되어야 한다. 현재의 초등교육에서는 한 교사가 여러 과목을 가르치는 체제이므로 모든 교육대학생들이 배울 수 있도록 전공 필수 과목으로 지정하여 미래의 소프트웨어 교육에 대비해야 한다.

넷째, 소프트웨어 제작 교육에 필요한 교재와 교구가 개발되어야 한다. 현재 개발된 교재들은 대부분 컴퓨터교육이나 공학을 전공하는 대학생을 위한 교재이다. 따라서 초등학생부터 가르치기 위해서는 컴퓨터의 개념과 원리를 초등학생이 쉽게 이해할 수 있는 교재가 개발되어야 하며, 그 원리를 바탕으로 스크래치와 로보틱스, KODU, LOGO, Turtle Roy 등을 이용하여 프로그램을 직접 작성할 수 있도록 교재와 교구 등을 개발해야 한다.

#### 참 고 문 헌

- [1] A. Cohen.(2014). Computer Science Education in Israel. Ministry of Education.
- [2] CAS(2013). Computing in the national curriculum: A guide for primary teachers. Computing At School.
- [3] D. Frost, A. Verno, D. Burkhart, M. Hutton, K. North.(2009). A Model Curriculum for K - 12 Computer Science Level I Objectives and Outlines. CSTA.
- [4] Department for Education(2013). National curriculum in England: computing programmes of study. <https://www.gov.uk/government/collections/national-curriculum>.
- [5] Heo, H. & Lim, K.(2011). Competency Modeling of Learner and Teacher in the 21st Century. Korea Education and Research Information Service.
- [6] HITSA(2014). Programming at Schools and Hobby Clubs. <http://www.innovatsioonikeskus.ee/en/programming-schools-and-hobby-clubs>.
- [7] ITLresearch(2014). 21st Century Learning Design. <http://www.itlresearch.com/itl-leap21>.
- [8] Kim, H. & Lee, S.(2013). Analysis of the Status of the Adapting ICT into Education and Information Science in the Developed Countries. Korea Education and Research Information Service.
- [9] Korea Ministry of Education(2000). Guidelines of Information and Communication Technology in Education.
- [10] Korea Mistry of Science, ICT and Future Planning(2013). The 5th National Information Master Plan in Korea(2013~2017).
- [11] NIPA(2012). 2012 Software Industry Analysis Report. National IT Industry Promotion Agency.
- [12] S. Iyer, M. Baru, V. Chitta, F. Khan, U. Vishwanathan(2010). Model Computer Science Curriculum for Schools. Department of Computer Science and Engineering Indian Institute of Technology Bombay.
- [13] Reaktor(2014). Lasten koodikoulu. <http://koodikoulu.fi>.



저 자 소 개



**정 영 식**

1996 춘천교육대학교  
수학교육학과  
(교육학학사)  
2001 한국교원대학교  
컴퓨터교육과  
(교육학석사)  
2004 한국교원대학교  
컴퓨터교육과  
(교육학박사)  
2004~2011 한국교육개발원  
연구위원  
2004~현재 전주교육대학교  
컴퓨터교육과 교수  
관심분야: 컴퓨터교육, 프로그래밍,  
이러닝  
e-mail: nurunso@jnue.kr



**김 철**

1997 전남대학교 대학원  
전산통계학과(이학박사)  
1998 University of Washington  
(객원교수)  
1992 현재 광주교육대학교  
컴퓨터교육과 교수  
관심분야: 인터넷자원관리,  
교육용콘텐츠,  
로봇활용교육,  
e-Learning  
e-mail: chkim@gnue.ac.kr