Journal of The Korea Association
of Information Education Vol. 19, No. 4,
December 2015, pp. 459-466

http://dx.doi.org/10.14352/jkaie.2015.19.4.459
© 2015 KAIE

초등학생을 위한 알고리즘 및 프로그래밍 교육과정 모델 개발

정영식

전주교육대학교 컴퓨터교육과

요 약

2015 개정 교육과정에 포함된 초등학교 소프트웨어 교육 내용은 17시간이라는 시수 제한으로 인해 그 내용이 충분하지 않다. 따라서 본 연구에서는 알고리즘과 프로그램 교육에 대한 교육과정 모델을 개발하였다. 이를 위해 국내외 SW 교육과정과, Code.org, Blockly Games, Entry와 같은 SW 교육 플랫폼을 분석한 후 알고리즘과 프로그래밍 교육에 대한 내용 체계와 성취 기준을 제시하였다. SW 교육 플랫폼의 분석 결과는 알고리즘의 이해, 프로그래밍의 실제, 프로그램의 평가 등 3가지로 구분하였고, 교육 내용 체계와 성취 기준은 7단계로 구분하였으며, 각 영역별로 14개의 성취 기준을 제시하였다. 또한, 학년에 상관없이 누구든지 하위 단계를 이해하면 상위 단계로 진급할 수 있도록 성취 기준을 무지개 색으로 구분하였다.

키워드 : 초등학교, 알고리즘, 프로그래밍, 소프트웨어 교육, 교육과정, 무지개

A Development of Algorithm and Programing Curriculum Model for Elementary School Students

Youngsik Jeong

Dept. of Computer Education, Jeonju National University of Education

ABSTRACT

The content of software education for elementary school students in the 2015 revisions to the national curriculum are not sufficient because class time dedicated to software education has been limited to 17 hours in fifth and sixth grades. In this study, I developed the algorithm and programming model for Korea. I analyzed domestic and international software education curricula as well as training platforms, such as Code.org, Blockly Games, and Entry. The suggested algorithm and programming framework is known as the Rainbow system, which is divided into 7 steps, 14 criteria, and 3 content areas--understanding the algorithm, the actual programming, and evaluation of the program. Using the Rainbow system, once students have completed a level they can be promoted to the next stage, regardless of their grade.

Keywords : Elementary School, Algorithm, Programming, Software Education, Curriculum, Rainbow

논문투고 : 2015-11-19
논문심사 : 2015-12-04
심사완료 : 2015-12-11

1. 서론

교육부는 2015년 문이과 통합 교육과정을 발표하였다. 주요 개정 내용을 살펴보면, 초등학교 5-6학년을 대상으로 실과 17시간을 활용하여 소프트웨어(이하 SW) 교육을 강화하기로 하였다[11]. 최근 미국이나 영국에서 기존의 ICT 활용 교육에서 벗어나, 문제 해결을 위한 컴퓨팅(computing) 교육을 강화하려는 취지와 일맥상통한다[1][15].

그러나 실과에 포함된 SW 교육 내용은 17시간이라는 제한된 시간에 운영할 수 있는 것으로만 편성되어 있어, 그 내용이 매우 빈약하며, 흥미와 체험 중심의 맛보기 교육으로 채워질 우려가 있다. 또한, SW교육은 SW 활용을 기반으로 교수학습활동이 이루어지므로 SW의 활용 교육 또한 매우 중요하다. 즉, 과거의 ICT 교육도 여전이 필요하다. 그러나 2015년에 개정한 실과 교육과정에서는 ICT 활용 교육이 모두 삭제되어 주춧돌을 빼다가 기왓장을 만든 꼴이 되어 버렸다.

이러한 문제점을 해결하기 위해 교육부는 SW교육운영지침을 발표하였으나, 이마저도 수업 시수가 명확하게 제시되지 않고 창의적 체험활동 시간과 교과 시간을 할애하여 지도하도록 하고 있어[15], 사실상 초등학교에서 SW교육운영지침을 적용하는 데에는 한계가 많다.

내용 요소 측면에서, 새롭게 시작되는 SW교육과 기존의 ICT 교육과의 가장 큰 차이는 알고리즘과 프로그래밍 교육의 포함 여부이다. 2005년 정보통신기술교육 운영지침에서도 '정보처리의 이해' 영역에 알고리즘과 관련된 내용을 포함하였으나, 그 내용이 구체적이지 않았다[9][10].

알고리즘은 현대 사회에서 직면할 수 있는 다양한 문제를 해결하고자 할 때, 그 문제의 본질을 이해하고 접근할 수 있기 때문에 문제 해결 능력을 향상시키고[7], 긍정적인 태도를 길러주며, 프로그래밍 교육은 분석력, 논리력, 창의력을 함양시킬 수 있다[6]. 따라서 알고리즘과 프로그래밍 교육은 컴퓨터 과학의 기본 원리를 적용하여 일상생활에서 발생할 수 있는 문제를 해결할 수 있고, 미래 지식 정보사회가 요구하는 창의적인 인재를 육성하는 데 꼭 필요한 교육이다[14][16].

본 연구에서는 초등학교에서의 SW 교육을 강화시키기 위해 알고리즘과 프로그램 교육에 대한 교육과정 모델을 개발하였다. 이를 위해 국내외 SW 교육과정과 SW 교육 플랫폼을 분석한 후 알고리즘과 프로그래밍 교육에 대한 내용 체계와 성취 기준을 제시하였다.

2. SW 교육과정에 대한 고찰

SW교육은 학생들이 컴퓨터 과학적 사고(Computational thinking)와 창의적인 사고를 통하여 세상을 이해하고 변화시키도록 하는 것이다. 컴퓨팅은 수학, 과학, 디자인, 기술 등과 연관이 많고, 인공 시스템을 만들 수 있다. 컴퓨팅의 핵심은 컴퓨터 과학(Computer Science)이고, 이는 학생들에게 컴퓨터 원리와 계산 원리를 가르치고, 디지털시스템이 어떻게 동작되고, 프로그래밍을 통해서 지식이 어떻게 이용되는지를 가르치는 것이다[4]. 우리나라의 SW교육은 SW 활용 중심의 교육과 SW 제작 중심의 교육으로 구분할 수 있다.

2.1 SW 활용 중심의 교육

우리나라 교육과정에서 SW 교육은 1971년 문교부령 제286호에 의해 개정된 전문계 고등학교 교육과정 중 전기과와 전자과에 개설된 '전자계산기'의 지도 내용에서 처음 제시되었다. 전자계산기 과목은 전자계산기의 원리·구성·기능 등에 관한 사항과 기초적인 프로그래밍 언어, 프로그래밍 기법에 관한 개념을 가르쳤다. 특히 SW 교육과 직접적으로 관련 있는 프로그래밍 교육은 기계어, 어셈블리어, 컴파일러를 중심으로 이루어졌다[17].

7차 교육과정 이전까지는 프로그래밍 교육이 실시된 적이 있었으나, 이후에는 SW 제작보다는 활용 중심의 교육으로 전환되었다. 2000에 시행되었던 '정보통신기술 교육 운영 지침'에서 재량활동 2시간 중에서 1시간을 의무적으로 ICT 교육을 하도록 하였다. 지침의 내용은 정보의 이해와 윤리, 컴퓨터 기초, SW의 활용, 컴퓨터 통신, 종합 활동 등 5가지로 구성되어 있어 대부분 ICT 활용 교육에 치중하였다. 이러한 문제점을 해결하기 위해 2005년에 정보통신기술교육 운영 지침을 수정하여 정보 처리의 이해 영역이 추가되었으나, 2007개정 교육과정이 시행되면서 창의적 체험 활동 시간이 축소되어 의무적으로 시행하던 1시간이 사실상 폐지됨에 따라 대다수의 초등학교에서 ICT 교육마저도 시행하지 못하고 있다.

2.2 SW 제작 중심의 교육

21세기 학습자 핵심 역량을 키우기 위해서는 SW를 활용하는 수준에서 벗어나 제작하는 능력을 키워야 한다. 더 이상 SW의 Player가 아니라 SW의 Maker를 양성해야 한다.

이를 위해 교육부는 2015년 9월에 교육과정을 개정하면서 SW 교육을 강화시켰다. 초등학교 5-6학년을 대상으로 실과 17시간을 활용하여 SW의 이해, 절차적 문제해결, 프로그래밍 요소와 구조 등 SW 교육을 실시하기로 하였다. 그러나 SW교육 영역이 별도의 교육 분야로 편성되지 못하고, 기존의 '기술 시스템' 영역에 억지스럽게 추가되어, 핵심 개념과 맞지 않고, '일반화된 지식' 역시 SW 교육과는 전혀 상관없는 내용이 제시되고 있다.

또한, 2015 개정 교육과정이 시행되기 이전부터 SW 교육을 강화하기 위해 SW 교육 운영 지침을 발표하였는데, 그 주요 교육 내용으로 SW와 생활의 변화, 정보윤리, 문제 해결 과정의 체험, 알고리즘의 체험, 프로그래밍 체험 등을 포함하였다[12]. 그러나 이 운영 지침은 2005년 정보통신기술교육 운영지침과 같이 SW교육에 필요한 별도의 시수를 명시하지 않고, 창의적 체험활동 시간과 교과 시간을 활용하도록 하여 실제적으로 SW 교육이 학교에서 진행될지는 미지수이다.

3. 프로그래밍 플랫폼 분석

초등학생과 같이 어린 학생들에게 알고리즘과 프로그래밍 교육을 체계적으로 가르치고 있는 SW교육 플랫폼에는 Code.org와 Entry, Blockly Games 등이 있다[2][3][5]. 이들의 기능을 분석한 결과 다음과 같이 알고리즘의 이해, 프로그래밍의 실제, 프로그램의 평가 등 3가지 영역으로 구분할 수 있었다.

3.1 알고리즘의 이해

SW 플랫폼에서 설명하고 있는 알고리즘 이해 과정의 주요 내용은 다음과 같다.

첫째, 일상생활에서 일어나는 일은 순서가 있음을 이해하고, 다양한 명령을 몸짓이나 손짓으로 표현하거나, 말과 글로 표현하는 명령어에 대해 이해하도록 한다. 또한, 엘리베이터에서 층 버튼 누르기, 자판기 메뉴 버튼 누르기 등과 같이 생활 속의 알고리즘을 찾아 설명한다.

둘째, 주어진 명령어들 중에서 반복적으로 사용되는 규칙을 찾아 그것을 줄여서 표현하고, 일상생활에서 사용하는 다양한 프로그램의 기능을 이해하며, 컴퓨터를 작동시키는 데 필요한 SW의 개념과 역할을 이해시킨다.

셋째, 복잡한 문제를 해결하기 위해 문제 해결 과정을 문장, 그림, 기호, 순서도 등으로 표현하고, 그것을 프로그램으로 작성한다. 문제 해결을 위해 불필요한 요소를 제거하고, 해결 방안을 설계하고, 프로그래밍 과정을 통해 SW로 구현하여 자동화할 수 있는 능력을 기른다.

넷째, 쓰임에 따라 다양한 프로그래밍 언어가 존재하며, 블록 프로그래밍 언어를 사용하는 데 필요한 SW를 설치하여 간단한 프로그램을 작성한다. 또한, 정렬 알고리즘과 검색 알고리즘의 개념을 이해한다.

3.2 프로그래밍의 실제

프로그래밍의 실제는 Code.org와 Blockly Games, Entry에서 제시한 여러 가지 미션 등을 분석한 후 초등학교 프로그래밍 교육의 범위에 포함될 수 있는 영역을 추출하였다. 그 결과 순차문, 반복문, 조건문, 자료와 연산, 함수와 이벤트 등으로 구분할 수 있었다.

첫째, 순차문은 이동, 회전, 점프 등의 개념을 포함하였다. 이동은 단순 이동에서부터 시작하여 상하좌우의 방향 이동, 일정한 거리만큼 이동하는 거리 이동으로 구분된다. 회전은 좌회전과 우회전에서 시작하여 주어진 각도를 선택하여 회전하거나 사용자가 특정한 각도를 입력하여 회전하는 것을 포함하였다. 점프는 장애물을 뛰어넘는 것부터 시작하여 일정한 거리를 뛰어넘는 내용으로 발전하였다.

둘째, 반복문에서는 단순 반복, 중첩 반복, 도형 그리기, 패턴 그리기를 포함하였다. 반복 구조는 크게 단순 반복과 중첩 반복으로 구분하였고, 반복의 종류는 repeat, until, while, for 등을 다뤘다.

셋째, 조건문의 핵심 개념은 산술 조건, 상황 조건, 단순 조건문, 중첩 조건문, 단방향 조건문, 양방향 조건문을 포함하였다.

넷째, 자료와 연산의 핵심 개념은 변수, 배열, 리스트, 자료형, 연산자, 입출력 등을 포함하였다. 배열은 동일한 형태의 변수를 여러 개 사용할 때 불편함을 해소하고, 변수의 접근을 자동화하기 위해 배열이 필요함으로 알게 하되, 1차원 배열과 리스트를 연계하여 학습하였다. 자료형은 숫자형, 문자형, 문자열 등 3가지 형태만을 포함하였고, 연산자는 산술 연산자, 비교 연산자, 논리 연산자를 포함하였다. 입출력은 키보드와 모니터를 활용한 기본 입출력과, 파일 입출력을 포함하였다.

다섯째, 함수와 이벤트에서는 함수, 이벤트, 매개 변수를 포함하였다. 객체와 객체 간에 메시지를 주고받을 때 전역변수보다는 이벤트를 사용하도록 하였고, 이벤트는 장치 이벤트와 상황 이벤트로 구분하였다.

3.3 프로그램의 평가

프로그램의 평가는 한국정보교육학회에서 개발한 SW 교육과정의 내용과 Code.org의 각 미션에서 제시한 내용 중에서 핵심 개념을 추출하여 단계적으로 제시하였다. 프로그램 평가와 관련된 핵심 개념은 오류 수정과 성능 개선으로 구분하였다.

첫째, 오류 수정은 주어진 구문에서 오류를 찾아 발견하고, 그것을 수정하는 것을 포함하였다. 수정은 특정 구문을 추가하거나 제거하기, 순서를 변경하기, 입력 값을 수정하기 등의 내용을 포함하였다.

둘째, 성능 개선은 오류는 없지만 구문이 복잡하거나 실행 시간이 오래 걸리는 구문을 수정하는 것이다. 순차문 중에서 반복적으로 사용되는 것을 반복문으로 개선하는 내용을 포함하였다. 또한, 반복문과 순차문이 반복적으로 사용될 때 함수문으로 개선하는 내용을 포함하였으며, 복잡한 알고리즘을 단순화시키거나 시간이 오래 걸리는 알고리즘을 개선하도록 하였다.

4. 교육과정 모델 개발

본 연구에서는 2014년에 개발된 한국정보교육학회에서는 알고리즘 분야와 프로그래밍 분야를 소프트웨어 제작 영역에 포함하여 초등학교 1학년부터 중학교 3학년까지의 과정을 각각 4가지 영역으로 구분하여 제시하였다[4].

첫째, 알고리즘 분야의 핵심 내용은 생활과 알고리즘, 알고리즘의 표현, 간단한 데이터 구조, 다양한 알고리즘과 데이터 구조 생성으로 구성하였다.

둘째, 프로그래밍 분야의 핵심 내용은 프로그램의 세계, 프로그램의 이해와 간단한 프로그램 제작, 프로그래밍 개발 환경과 다양한 프로그램 제작, 소프트웨어 제작 관리를 포함하였다.

4.1 알고리즘 분야의 성취 기준

알고리즘 분야에서는 알고리즘의 의미를 이해하고, 문제를 해결하기 위해 선택 구조와 반복 구조의 알고리즘을 작성할 수 있으며, 알고리즘의 문제점을 수정하여 보다 효과적인 알고리즘으로 개선할 수 있다. 알고리즘 영역에 대한 구체적인 성취 기준은 다음과 같다.

(빨강)

1. 일의 순서에 따라 그림이나 문장을 나열한다. 일상생활에서 발생할 수 있는 일을 그림 카드나 짧은 문장 카드로 제시하면, 일이 일어난 순서에 따라 그림 카드나 짧은 문장 카드를 나열하고, 그것을 간단하게 말로 설명한다.

2. 일상생활의 활동을 순서에 따라 표현한다. 일상생활에서 발생할 수 있는 일을 순서에 따라 간단한 말이나 글로 표현한다. 일이 일어난 순서를 간단한 문장으로 표현하고, 그것을 말로 설명한다. 또한, 짧은 문장을 그림으로 표현하여 말과 글, 그림으로 표현했을 때의 편리한 점과 불편한 점을 말한다.

(주황)

3. 문제 해결 과정을 순서에 따라 표현한다. 일상생활에서 발생하는 문제를 해결하기 위해 문제를 이해하고, 해결 방법을 순서에 따라 간단하게 말이나 글로 표현한다. 문제 해결하기 위해 중요한 것과 덜 중요한 것, 필요한 것과 불필요한 것을 구분한다.

4. 알고리즘은 일이 일어난 순서임을 이해한다. 문제를 해결하기 위해 필요한 일을 순서대로 나열한 것이

알고리즘을 이해한다. 문제를 해결하는 위한 과정에는 여러 가지가 있으므로, 특정 문제를 해결하기 위한 알고리즘도 여러 가지가 있을 수 있음을 이해한다.

(노랑)

5. 문제해결 과정을 그림이나 기호로 표현한다. 문제해결 과정을 말이나 글로 표현하는 것과, 그림이나 기호로 표현했을 때의 장점과 단점을 이해한다. 그림보다는 기호로 표현했을 때 보다 간단히 표현할 수 있음을 이해하고, 여러 가지 기호를 이용하여 문제 해결 과정을 표현한다.

6. 문제해결 과정을 순서도로 표현한다. 문제 해결 과정을 스스로 만든 기호와 약속된 기호로 표현했을 때의 장점과 단점을 말한다. 순서도는 알고리즘을 표현하는 약속된 기호임을 이해하고, 순서도에서 사용되는 여러 가지 기호를 이해하고, 용도에 맞게 사용한다.

(초록)

7. 반복 구조를 순서도로 표현한다. 일상생활에서 반복적으로 일어나는 일을 순서도로 간단히 표현한다. 반복적으로 일어나는 일을 순차 구조로 표현할 때의 불편한 점을 이해하고, 가급적 반복 구조로 표현하려는 태도를 갖는다.

8. 선택 구조를 순서도로 표현한다. 조건에 따라 문제를 해결하는 방법이 달라질 수 있음을 이해하고, 선택 구조를 순서도로 표현한다. 조건을 만족했는지를 정확히 판단하며, 조건을 만족하지 못했을 때 일을 처리하는 방법을 순서도로 표현한다.

(파랑)

9. 컴퓨터의 기능과 알고리즘의 관계를 이해한다. 컴퓨터는 여러 가지 알고리즘에 의해 작동됨을 이해하고, 그 알고리즘은 명령어로 구성되어 있음을 이해한다. 컴퓨터의 간단한 기능을 알고리즘으로 표현하고, 그것을 설명한다.

10. 알고리즘의 작동 결과를 예측한다. 문제를 해결하기 위한 알고리즘의 실행 과정을 이해하고, 그 결과를 예측한다. 또한, 주어진 조건에 따라 결과가 달라지는 알고리즘을 찾아내고, 알고리즘의 작동 결과와, 결과가 달라지는 이유를 설명한다.

(남색)

11. 여러 알고리즘의 장단점을 비교한다. 문제를 해결하기 위해서 다양한 알고리즘을 만들고, 만들어진 알고리즘의 장점과 단점을 말한다. 또한, 알고리즘을 비교하는 기준을 만들고, 그에 따라 공통점과 차이점을 설명한다.

12. 알고리즘의 오류를 찾아 수정한다. 주어진 알고리즘의 결과를 예측하고, 잘못된 결과가 발생했을 때 그 이유를 설명한다. 또한, 문제를 해결하기 위해 알고리즘을 만들고, 잘못된 부분이 있으면 찾아서 수정한다.

(보라)

13. 가장 효과적인 알고리즘을 찾는다. 여러 사람이 만든 알고리즘을 특정 기준에 따라 비교하며 설명한다. 명령의 수행 횟수나 처리 속도 등을 고려하여 알고리즘을 비교하고, 주어진 알고리즘에서 문제를 해결하는 데 가장 효과적인 알고리즘을 찾는다.

14. 보다 효과적인 알고리즘으로 수정한다. 일상생활의 문제를 해결하기 위해 학생 스스로 알고리즘을 만들 수 있다. 자신이 만든 알고리즘을 다른 사람과 공유하며 의견을 주고받을 수 있고, 자신의 알고리즘이 갖고 있는 오류나 문제점을 찾아 개선할 수 있다.

4.2 프로그래밍 분야의 성취 기준

프로그래밍 분야에서는 일상생활의 문제를 해결하기 위해 간단한 프로그래밍 도구를 활용하여 프로그램을 작성한다. 알고리즘의 순차 구조와 반복 구조, 선택 구조를 프로그램으로 작성할 수 있으며, 여러 가지 변수와 연산자, 함수, 이벤트 등을 이용하여 자료를 처리할 수 있는 프로그램을 작성할 수 있다. 프로그래밍 영역에 대한 구체적인 성취 기준은 다음과 같다.

(빨강)

1. 간단한 명령어를 이용하여 물체를 이동하거나 회

전한다. 앞으로 이동, 뒤로 이동, 오른쪽으로 이동, 왼쪽으로 이동, 위로 이동, 아래로 이동과 같은 여러 가지 방향을 지시하는 명령어를 이용하여 물체를 이동할 수 있다. 또한, 오른쪽으로 회전, 왼쪽으로 회전과 같은 명령어를 이용하여 물체를 회전시킬 수 있다.

2. 간단한 명령어를 이용하여 간단한 도형을 그린다. 일을 처리할 때에는 2개의 이상의 명령이 동시에 이루질 수 있음을 이해한다. 이동 또는 회전하는 명령과 동시에 그리는 명령을 함께 사용하여 직사각형이나 정사각형과 같은 간단한 도형을 그릴 수 있다.

(주황)

3. 정보기기는 프로그램에 의해 작동된다는 것을 이해한다. 일상생활에서 사용하는 정보기기는 프로그램에 의해 작동되고, 그 프로그램은 여러 가지 명령어로 구성되어 있음을 이해한다. 또한, 프로그램을 작성하는 과정을 프로그래밍이라고 하며, 이러한 프로그래밍은 여러 가지 SW를 통해 작성할 수 있음을 이해한다.

4. 일상생활에서 사용하는 프로그램의 역할을 이해한다. 일상생활에서 사용하는 프로그램의 기능과 역할을 이해할 수 있으며, 간단한 기준에 따라 프로그램을 구분할 수 있다. 프로그램이 없을 때의 불편한 점을 이해함으로써 프로그램의 중요성에 대한 인식한다.

(노랑)

5. 프로그래밍 도구를 활용하여 간단한 프로그램을 작성한다. 그래픽 기반의 프로그래밍 도구를 이용하여 간단한 프로그램을 작성할 수 있다. 간단한 명령어를 이용하여 삼각형, 사각형, 다각형, 원 등 다양한 도형을 그릴 수 있다.

6. 변수와 산술 연산자를 이해하고 사용한다. 변수를 이용하여 외부기기를 통해 입력된 값을 저장할 수 있음을 이해하고, 변수를 활용한 프로그램을 작성할 수 있다. 이때 전역 변수와 지역 변수의 차이를 이해하고, 목적에 맞게 변수를 만들어 사용할 수 있다. 또한, 사칙연산자와 같은 간단한 산술 연산자를 이용하여 변수의 값을 변경하는 프로그램을 작성할 수 있다.

(초록)

7. 반복문의 필요성을 이해하고 사용한다. 일정한 횟수만큼 반복되는 구문이 있을 경우 그것을 반복문으로 표현할 수 있다. 반복되는 횟수가 많을 경우 그것을 순차문으로만 구성했을 때의 불편한 점을 이해하고, 반복되는 순차문을 반복문으로 수정하는 방법을 알고 사용한다.

8. 비교 연산자를 이용하여 간단한 조건문을 사용한다. 같다(=). 크다(>), 작다(<), 크거나 같다(>=), 작거나 같다(<=), 같지 않다(<>) 등 다양한 비교 연산자를 이용하여 수식의 크기를 비교할 있다. 비교 연산자의 결과에 따라 조건의 만족 여부를 판단하고, 조건을 만족했을 때에만 특정 구문을 수행하는 간단한 조건문을 이용하여 프로그램을 작성할 수 있다.

(파랑)

9. 여러 형태의 반복문과 조건문을 이해하고 사용한다. 일정 회수만큼 반복하기, 무한 반복하기, 조건을 만족할 때까지 반복하기 등 다양한 형태의 반복문의 사용법을 이해하고 사용할 수 있다. 또한, 특정 조건을 만족할 때와 그렇지 않을 때를 구분하여 처리할 수 있는 조건문을 사용할 수 있다.

10. 간단한 함수와 이벤트를 이해하고 사용한다. 화면 입출력이나 랜덤 함수 등 간단한 내장 함수를 호출하여 프로그램을 작성할 수 있다. 또한, 이벤트를 만들어 객체와 객체 사이에 메시지를 전달하는 방법을 이해하고 사용한다.

(남색)

11. 논리 연산자를 이해하고 사용한다. 그리고(and), 또는(or), 부정(not) 등의 논리 연산의 결과를 이해하고 논리 연산자를 사용할 수 있다. 또한, 조건문에서 조건에 따라 여러 번 분기되는 것을 논리 연산자를 이용하여 간단하게 표현한다.

12. 반복문과 조건문을 중첩하여 사용한다. 반복문 안에서 새로운 반복문을 만들거나, 조건문 만에서 새로운 조건문을 만들어 사용한다. 또한, 반복문 안에 조건

문을 사용하거나, 조건문 안에 반복문을 사용하는 등 다양한 형태의 중첩문을 사용한다.

(보라)

13. 필요한 함수를 만들어 사용한다. 순차문, 반복문, 조건문 등을 포함한 구문이 반복적으로 사용되는 경우, 또는 특정 기능을 수행하는 경우에 그것을 함수로 정의하고 호출한다. 또한, 함수의 매개변수의 개념과 사용법을 이해하고 사용한다.

14. 여러 가지 형태의 자료를 파일에 저장하고 읽는다. 동일한 자료 형태의 변수가 반복적으로 사용될 때 그것을 배열로 수정하여 사용하며, 배열에 포함된 특정 값을 인덱스와 반복문을 이용하여 접근한다. 또한, 변수나 배열에 저장된 값을 파일에 저장하며, 파일에 저장된 값을 불러온다.

5. 결론 및 제언

본 연구에서 제안한 알고리즘과 프로그래밍에 대한 교육과정 모델은 총 7단계로 구성되어 있으며, 유치원과 초등학교에서 활용할 수 있다. 7단계를 무지개 색으로 제시한 것은 태권도 품띠와 같이 학생들이 학년의 구분 없이 자유롭게 알고리즘과 프로그래밍 능력을 키워 상급 단계로 올라갈 수 있도록 하기 위함이다. 제시된 성취 기준이 실제 학교 현장에서 사용되기 위해서는 다음과 같은 노력이 필요하다.

첫째, 단계별로 알고리즘과 프로그래밍 교육에 필요한 구체적인 평가 기준과 함께 교재와 교구를 개발해야 한다. 본 연구에서 제시한 성취기준만으로는 학교 현장에서 무엇을 어떻게 가르쳐야 할지 알기 어렵다. 따라서 제시한 성취 기준을 달성하기 위해 필요한 다양한 교재와 교구를 개발한다면 현장 교사들이 학생들을 가르치는 데 큰 도움이 될 것이다. 따라서 이에 대한 후속 연구가 필요하다.

둘째, 2015 개정 교육과정에 제시된 초등 실과의 SW 교육 내용은 매우 빈약하다. 17시간이라는 제한된 시간만으로 알고리즘 교육과 프로그래밍 교육을 하는 데에는 한계가 있다. 따라서 본 연구에서 제시한 교육 내용을 초등학교에서 학습할 수 있도록 창의적 체험활동 시간 등을 활용하여 충분한 수업 시수를 확보해야 한다.

셋째, 알고리즘과 프로그래밍 교육을 실제 적용하는 데 필요한 다양한 교수학습 방법과 평가 방법을 제시하고 그에 따른 구체적인 수업 사례와 우수 사례들 발굴해야 한다. 예를 들면, 창의적 체험활동 시간이나 방과 후 활동 시간에 본 연구에서 제시한 교육과정을 적용한다면 확보된 시수에 따른 가장 적합한 내용 요소를 추출하여 그것을 기반으로 교수학습지도안을 작성한 후 교실에서 적용하고 개선 방안을 마련해야 한다.

넷째, 알고리즘과 프로그래밍 교육에 필요한 교원을 양성해야 한다. 교사 연수 이외에도 초등교원 양성 단계에서부터 체계적으로 배울 수 있도록, 교육대학교의 교육과정에 알고리즘과 프로그래밍 교육에 대한 강좌를 개설해야 한다. 초등학교에서의 교과는 모든 교사가 가르쳐야 하므로, SW교육 역시 모든 교사가 가르쳐야 한다. 따라서 교육대학교의 모든 과 학생들이 이수할 수 있도록, 알고리즘과 프로그램에 대한 과목을 개설하여 전공 필수로 운영되어야 한다.

알고리즘과 프로그래밍 교육은 미래의 SW 중심 사회에서 살아가는 데 필요한 컴퓨터 과학적 사고를 키우는 데 필수적이다. 따라서 모든 초등학생들이 알고리즘과 프로그래밍 교육을 받을 수 있도록, 이번 연구 결과를 토대로 좀 더 개선된 SW 교육과정이 만들어지길 기대한다.

참고문헌

[1] CAS (2013). Computing in the national curriculum: A guide for primary teachers. Computing At School.
[2] Code.org (2015). Hour of Code. http://www.code.org.
[3] Entry Education Research(2015). Entry. http://play-entry.com.
[4] Gapsu Kim, Chul Kim, Hyunbae Kim, Ingi Jeong, Youngsik Joeng, Sunghun Ahn, Jongwoo, Kim (2014). A Study on Contents of Information Science Curriculum. *Journal of The Korean association of information education, 18*(1), 161-171. Korean Association of Information Education.
[5] Google (2015). Blockly Games. https://block-

ly-games.appspot.com.
[6] Gyo-Sik Moon (2007). On the Direction of the Computer Algorithm Education Based on Conceptual Algorithms. *Journal of The Korean association of information education, 11*(1), 29-38. Korean Association of Information Education.
[7] Hong-rae Kim, Seung-Jin Lee (2013). Analysis of ICT Education in Advanced Country. Korean Education and Research Information Service.
[8] Hwa-Kyong Lim, Jin-Ah Kim, Jin-Ho Bae (2004). A Instructional Learning of Sort Algorithm by Role-play for Computer Education in the Primary School. *Jounal of the Korean Institute of Information Scienctists and Engineers, 4.*
[9] Ministry of Education (2000). Guidelines of Information and Communication Technology in Education Amendments.
[10] Ministry of Education (2005a). Guidelines of Information and Communication Technology in Education Amendmentss.
[11] Ministry of Education (2015b). The revised national curriculum 2015 for Primary and Secondary Schools.
[12] Ministry of Education (2015c). Guidelines of Software Education Management.
[13] Min-Young Lim (2006). A study on learnability of search and sort algorithm of data structure in elementary school computer education. Jinju National University of Education.
[14] Ok-Kyoung Han, Jae-Hyun Kim (2009). A Study on a Strategy for Spiral Curriculum Model in Algorithm Education. *Proceeding of the Korean association of computer education, 13*(1), 129-134. Korean association of computer education.
[15] Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C. and Verno, A. (2002). A Model Curriculum for K-12 Computer Science. Report of the ACM K-12 Education Task Force Computer Science Curriculum Committee.
[16] Youngsik Jeong (2014). A Study on the Content Framework of Algorithm Education in Primary and Middle School. *Journal of The Korean association of information education, 18*(2), 275-284. Korean Association of Information Education.
[17] Youngsik Jeong, Jungsu Yu, Jinsuk Lim, Yukyoung Son (2015). Theory of Software Education. Simass.

저자소개

정 영 식

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