

2015 개정 교육과정의 소프트웨어 교육에 대한 초등 교사들의 인식 분석

김갑수

서울교육대학교 컴퓨터교육과

요 약

21세기 지식 정보 사회의 창의적인 인재 양성을 위해 2018년도부터 초등학교에서 소프트웨어 교육을 실시한다. 2015년에는 초등학교에서 실시되는 소프트웨어 교육의 교육 내용과 성취기준을 만들었다. 본 연구는 초등학교 교사 199명을 대상으로 소프트웨어 교육에 대한 인식을 조사했고, 조사 결과는 다음과 같다. 첫째, 소프트웨어 교육이 초등학교에 필요하고, 17시간으로는 부족하다. 둘째, 현재 교사들은 일반적인 소프트웨어 교육의 교육 내용, 교수 학습 방법, 평가 방법, 자료 개발 방법 등의 내용을 잘 모른다. 셋째, 성취기준에 대한 이해, 성취기준 적합한 교육 자료 개발과 지도 방법에 대해서도 인지가 낮다. 따라서 초등학교 모든 교사가 교사 연수를 통하여 성취기준에 적합한 교수학습 자료 개발, 평가 방법, 지도 방법을 알 수 있게 해야 한다.

키워드 : 초등학생, 소프트웨어 교육, 소프트웨어 교육과정, 내용 체계, 성취기준

A Recognition Analysis of Elementary Teachers for Software Education of 2015 Revised Korea Curriculum

Kapsu Kim

Dept. of Computer Education, Seoul National University of Education

ABSTRACT

In order to nurture creative talent in the 21st century knowledge-based society in elementary education software is carried from the year 2018. The educational content and achievement standards to conduct a software education had been made in the 2015. In this study, the recognition of educational software for elementary school teachers 199 people is investigated. Findings are as follows. Elementary education is the first software required, and is lacking in 17 hours. Second, the idea of a common training software education, teaching and learning methods, evaluation methods, how to develop information materials. Third, lower cognitive development and educational materials for the teaching methods appropriate for understanding, achievement standards for achievement standards. Therefore, should allow teacher training teaching materials development, assessment methods, teaching methods suitable for the achievement standards available to all teachers throughout the school know.

Keywords : Elementary Student, Software Education, Software Curriculum, Contents, Achievement

2015년도 서울교육대학교 교내연구비에 의하여 연구되었음.

논문투고 : 2015-12-31

논문심사 : 2015-12-31

심사완료 : 2016-02-12

1. 서론

21세기 지식 정보 사회에서는 창의성을 갖고 새로운 것을 만들어내는 창조 인재가 필요하다[1]. 전 세계는 무한경쟁의 시대로 접어들었고, 학생들은 지식정보를 생성하여 국제적으로 유통시키고 있다. 이에 따라 대부분의 국가들이 산업 사회의 교육 방법이 아니라 지식 정보 사회의 교육 방법으로 컴퓨터 교육을 체계적으로 실시하고 있다.

영국·미국·독일 등은 산업사회에 수학 및 과학을 기본 교과로 하여 공교육 틀을 유지해 오다가 21세기부터 본격적인 지식 정보 사회의 시작으로 정보를 단순히 활동하는 것이 아닌, 정보를 이용하여 새로운 것을 창조하기 위해서 컴퓨터 교육을 중요하게 다루고 있다.

먼저 영국은 매우 선도적으로 컴퓨터를 활용하는 교육에서 학생들이 직접 소프트웨어를 제작하기 위한 교육을 실시하고 있다. 2014년부터 초등학교 1학년부터 컴퓨팅이라는 교과목을 신설하여 실시하고 있다[2][3]. 영국 런던의 과학박물관에 가면 수학과 더불어 컴퓨팅이라는 구역(zone)을 만들어 학생들에게 관심을 가지게 하고 있다.

미국의 경우에는 ISTE와 ACM 기관에서 컴퓨터 교육에 대한 표준안을 만들어 소프트웨어 교육 과정을 만들고 있다. ACM의 컴퓨터 교사 모임에서 2011년부터 컴퓨터 교육의 표준안을 제정하여 전국의 교사들에게 권고하고 있다. 여기에는 기존의 컴퓨터 활용교육에서 사고력을 개발하는 계산 사고와 프로그래밍 영역을 강화하여 초등학교 1학년부터 권고 하고 있다[4][14].

독일의 경우에도 2008년부터 5학년부터 컴퓨터 교육을 실시할 것을 권고하는 표준안을 만들어서 각 주에서 컴퓨터 교육을 실시하고 있다[6].

우리나라 컴퓨터 교육은 교육과학기술부에서 2008년부터 정보통신기술운영 지침을 폐지하고 중학교에서는 정보 교과가 선택으로 변경이 되었다. PISA의 통계자료를 분석하면 현재 우리나라 학생들의 컴퓨터 능력은 현저히 떨어지고 있다[5][11][13].

이러한 심각한 컴퓨터 교육의 위기 상황에서 새로운 교육의 틀을 모색하기 위해서 2015년 9월에 문·이과 통합교육과정을 만들 때, 우리나라도 소프트웨어 교육을 강화하기 위해서 소프트웨어 교육과정을 발표하였다.

지금까지 컴퓨터 교육의 필요성을 한국정보교육학회, 한국컴퓨터교육학회 등이 제기해 온 결과 소프트웨어 교육과정을 만들기에 이르렀다[7][8][10].

2015년 교육부가 소프트웨어 교육과정을 고시하였고 2018년부터 초등학교에서부터 소프트웨어 교육을 실시할 예정이다[9][12]. 한국교육학술정보원은 소프트웨어 교육 운영지침을 만들어서 2018년부터 각 학교가 사정에 맞추어 소프트웨어 교육을 실시할 수 있는 가이드라인을 제시하였다[2]. 교육부는 2014년부터 소프트웨어 교육을 공론화하여 2015년 9월에 발표하였는데 초등학교의 경우 실과 교과에서 17시간 이상 소프트웨어 교육을 할 수 있게 소프트웨어 교육내용 체계와 성취기준 및 해설 등을 발표하였다[9][12].

2015년 9월에 발표한 소프트웨어 교육의 내용체계 및 성취기준에 대해서 초등학교 선생님들이 어느 정도 알고 있고, 가르칠 수 있는지에 대한 연구가 필요하다. 따라서 본 연구에서는 초등학교 현장 교사 199명을 대상으로 설문조사를 실시하여 2015년 개정 초등학교에서 실시할 소프트웨어 교육에 대한 인식조사를 하고, 교육부가 제시한 성취기준에 대한 수업 시수 등이 얼마나 타당한지를 살펴본다.

이런 현장 연구를 통해서 소프트웨어 교육이 초등학교에서 올바르게 정착될 수 있는 방안은 무엇인지 본 연구에서 찾아보고자 한다.

제2장에서 2015년 개정된 우리나라 소프트웨어 교육과정을 간단히 살펴보고, 제3장에서는 초등학교 소프트웨어 교육에 대한 내용체계와 성취기준에 대한 현장 교사들의 설문을 분석한 것을 설명하고, 제4장은 본 연구의 결론 및 제언이다.

2. 2015 소프트웨어 교육과정

2015년 개정 문·이과 통합형 교육과정에서 초등학교 실과 교과의 내용체계 중에서 컴퓨터 교육을 실시해야 하는 부분에서 소프트웨어 교육에 해당되는 부분을 추출하여 정리하면 다음 <Table 1>과 같이 구성할 수 있다. 실과 교과의 소프트웨어 교육 지식과 내용은 영역 부분, 핵심 개념, 일반화된 지식, 내용 요소 및 기능으로 나누었다. 소프트웨어 교육은 기술 시스템 영역의 한

부분으로 되어 있고, 기술 시스템 영역은 소문의 한 분야 되어 있다. 기술 시스템의 내용 요소로는 소프트웨어의 이해, 절차적 문제 해결, 프로그래밍 요소와 구조로 되어 있다.

이에 대한 구체적인 일반적인 지식과 내용 요소는 다음 <Table 1>과 같다.

<Table1> Software Education Knowledge and Contexts[12]

| area | general knowledge | contexts elements |
|--------------------|---|--|
| Technology systems | Communication techniques to the production, processing and sharing information by sending and receiving via a variety of means and devices. | Understanding the software. procedural Problem Solving. Programming elements and structures. |
| | | |

초등학교 실과 교과의 내용체계에 대한 구체적인 성취기준과 해설[12]은 다음과 같다.

[6실04-07] 소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다. 컴퓨터에 사용된 소프트웨어 이외에도 휴대폰, 가전제품, 사물인터넷 제품까지 여러 상황에서 사용된 소프트웨어를 탐색해 보고 우리 생활에 미치는 영향을 이해한다.

[6실04-08] 절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다. 절차적 사고란 문제를 효율적으로 해결하기 위해 문제를 작은 단위로 나누고, 각각의 문제를 단계별로 처리하는 사고 과정으로, 일상생활 속의 사례들을 찾아보고 절차적 사고 과정을 문제 해결에 적용한다.

[6실04-09] 프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다. 블록 기반의 교육용 프로그래밍 도구를 활용하여 기초적인 프로그래밍 과정을 체험 하고 자신만의 간단한 프로그램을 만들어 본다.

[6실04-10] 자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.

수치 값을 입력하여 덧셈이나 뺄셈의 결과를 출력하거나, 복수의 문자열을 입력하여 두 문자열을 서로 연결한 결과를 출력하는 프로그램을 만들어 봄으로써, 소

프트웨어의 입력, 처리, 출력 과정을 이해한다.

[6실04-11] 문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다. ‘순차’는 명령문을 위에서 아래로 하나씩 순차적으로 수행하는 과정이며, ‘선택’은 주어진 조건에 따라 명령문을 선택적으로 수행하는 과정이다. ‘반복’은 명령문을 특정 횟수만큼 반복하거나, 주어진 조건이 만족할 때까지 반복하는 과정이다. 일상의 문제를 해결하는 프로그램을 만드는 기초 과정을 통해 위 프로그램의 3가지 구조를 이해한다.

3. 설문 분석 및 논의

3.1 개요

2015년 문·이과 통합 교육과정을 만들 때 실과에 편제하고 독립적인 교과를 두지 않았다. 중학교에서는 정보라는 분야를 두고 있지만 초등학교와는 교과목 분류상 연계가 전혀 이루어지고 있지 않다. 이러한 상황에서 초등학교에서 소프트웨어 교육을 실시하기 위해서는 초등학교 교사들은 어떻게 인식하고 있는지 분석할 필요가 있다.

3.2 연구 문제

본 연구에서 연구하고자 하는 연구 문제는 다음과 같다. 첫 번째 문제는 초등학교 교사들이 소프트웨어 교육에 대해서 얼마나 알고 있고 필요성을 느끼는가에 대한 것이다.

두 번째 문제는 2015년도 교육부에서 고시한 소프트웨어 교육과정이 실과 교과에 있는 것이 타당한가에 대한 것이다.

세 번째 문제는 교사들이 소프트웨어 교육 내용, 교수·학습 방법, 평가 및 자료 개발에 대한 지식이 어느 정도 있는가에 대한 것이다.

네 번째 문제는 교사들이 2015년도 교육부에서 고시한 소프트웨어 교육과정에서 교육 내용체계에 대해서 얼마나 많이 알고 있는가에 대한 것이다.

다섯 번째 문제는 교사들이 2015년도 교육부에서 고시한 소프트웨어 교육과정에서 성취기준 및 해설에 대

해서 얼마나 많이 알고 있는가에 대한 것이다.

3.3 연구 대상 및 기간

본 연구는 서울 및 경기지역 초등학교 교사 199명을 대상으로 2016년 1월 2일부터 1월 7일까지 인터넷으로 설문조사한 것을 바탕으로 진행하였다. 교사들을 분류하면 컴퓨터 교육 전공 교사는 48명, 그 이외의 교사들은 151명이다.

3.4 연구 내용 및 분석

3.4.1 필요성

첫 번째 문제는 초등학교 교사들이 소프트웨어 교육에 대한 필요성을 얼마나 알고 있느냐에 대한 인식 조사이다. 조사 내용은 교사들이 2018년도부터 소프트웨어 교육을 초등학교에서 실시한다는 것을 아는가에 대한 질문과 미래의 학생들을 위해서 소프트웨어 교육이 초등학교에서 필요한가 하는 질문 및 소프트웨어가 우리 생활에 중요한 역할을 하는가 하는 질문이다.

이에 대한 질문은 다음과 같다.

- 1) 나는 2018년도부터 소프트웨어 교육을 초등학교에서 실시한다는 것을 알고 있다.
- 2) 나는 미래의 학생들을 위해서 소프트웨어 교육이 초등학교에서 필요하다고 생각한다.
- 3) 나는 소프트웨어가 우리 생활에 중요한 역할을 한다는 것을 알고 있다.

이 항목들에 대한 설문 결과는 다음 <Table 2>와 같다.

<Table 2> Necessity of Software Education

| items | not strongly agree | not agree | normal | agree | strongly agree |
|-------|--------------------|-----------|--------|-------|----------------|
| 1) | 25 | 32 | 19 | 54 | 69 |
| 2) | 4 | 14 | 38 | 74 | 69 |
| 3) | 2 | 5 | 21 | 87 | 84 |

<Table 2>의 설문 분석 결과는 다음과 같다. 먼저, “미래의 학생들을 위해서 소프트웨어 교육이

초등학교에서 필요하다.”에 대해 동의하지 않는 교사들이 18명으로 9%이고, 72%의 교사들은 초등학교에서 컴퓨터 교육이 꼭 필요함에 동의한다. 특히, 컴퓨터 교육을 전공한 79.2%의 교사들이 강력히 필요하다고 생각한다.

소프트웨어가 우리 생활에서 꼭 필요하다는 것을 인식하지 못하는 교사는 7명으로 3.5%이고, 86%는 우리 생활에서 소프트웨어가 매우 중요한 역할을 한다는 것을 인식하고 있다. 컴퓨터 교육을 전공한 초등학교 교사 94%가 미래 사회에 소프트웨어가 필요하다고 인식하고 있다. 2018년도부터 소프트웨어 교육을 실시한다는 것을 모르고 있는 초등학교 교사들이 29%이고, 잘 알고 있는 교사들은 35%이다.

여기서 소프트웨어가 앞으로 중요한 역할을 한다고 느끼는 교사들이 초등학교에서 소프트웨어 교육이 필요하다고 느끼는 교사들의 t검정을 해 보면 다음 <Table 3>과 같다.

<Table 3> Analysis on recognition of the SW education

| item | Avg | N | variance | t | Sig |
|-------|------|-----|----------|------|---------|
| 1) | 3.97 | 199 | 0.96 | 3.17 | .000*** |
| 2) | 4.25 | 199 | 0.62 | | |
| Total | | 398 | | | |

<Table 3>을 분석해 보면 소프트웨어가 중요하다고 느끼는 교사는 초등학교에서 소프트웨어 교육이 필요하다고 생각하는 것을 알 수 있다.

초등학교 교사들이 대부분 미래 사회에서 소프트웨어 교육의 중요성을 인식하고 있고, 초등학교에서 소프트웨어 교육이 필요하다고 인식하고 있다.

3.4.2 소프트웨어 교육이 성과에 있는 것이 타당한가?

소프트웨어 교육은 우리나라 교육에서 중요한 역할을 할 것 같지만 초등학교에서 독립교과로 구성되어 있지 않다. 중등에서는 정보라는 교과가 독립되어 있지만 초등학교에서는 정부정책상 독립 교과로 구성되어 있지 않다. 이에 대해 초등학교 교사들은 어떤 생각을 하고 있는지 분석하는 것은 중요하다. 따라서 이번 설문에서

는 이에 대한 설문 문항을 구성하여 <Table 4>와 같은 결과를 얻었다.

설문 문항은 다음과 같다.

- 1) 나는 소프트웨어 교육의 내용을 알고 있다.
- 2) 나는 소프트웨어 교육은 실과 교과에서 실시하는 것이 타당하다고 생각한다.
- 3) 나는 소프트웨어 교육을 위한 시간이 17시간으로 충분하다고 생각한다.

<Table 4> Validity of Software education

| items | not strongly agree | not agree | normal | agree | strongly agree |
|-------|--------------------|-----------|--------|-------|----------------|
| 1) | 23 | 48 | 55 | 49 | 24 |
| 2) | 16 | 37 | 46 | 71 | 29 |
| 3) | 31 | 50 | 52 | 56 | 10 |

<Table 4>의 내용을 분석하여 보면 초등학교 실과 교과에서 소프트웨어 교육을 실시해야 한다고 답한 교사는 50%이고, 그렇지 않는 교사가 50% 정도이다. 소프트웨어 교육 17시간은 충분하지는 않지만 만족하는 교사는 33%이고, 부족하다고 생각하는 교사가 41%이고, 나머지는 보통이라고 답했다. 이를 통해 초등학교 교사들은 17시간으로 소프트웨어 교육을 한다는 것에 대해 부족하다고 인식하는 것을 알 수 있다. 또한 소프트웨어 교육이 실과 교과에서 실시되는 것이 타당함에 대해서는 50% 정도가 찬성하지 않기 때문에 소프트웨어 교육을 독립적으로 편성해야 할 것이다. 이를 좀 더 자세히 분석하여 보면 컴퓨터 교육을 심화과정으로 전공한 교사들의 33%가 소프트웨어 교육이 실과에 있어도 된다고 하였고, 50%는 실과에 있으면 안 된다고 답하였다. 17%는 유보적인 입장을 취하였다.

현재 소프트웨어 교육 내용을 알고 있는 교사들은 37%에 불과하였다. 소프트웨어 교육 내용을 알고 있는 교사가 소프트웨어 교육을 실과에서 해야 한다는 것을 t검정을 해 보면 다음과 같다.

<Table 5> Analysis on recognition of the SW validity policy

| item | Avg | N | variance | t | Sig |
|-------|------|-----|----------|------|---------|
| 1) | 3.02 | 199 | 1.44 | 2.41 | 0.01*** |
| 2) | 3.30 | 199 | 1.36 | | |
| Total | | 398 | | | |

두 변수를 t검정을 통해 분석해 보면 두 집단이 유의 수준 1%에서 차이가 있다는 것을 알 수 있다. 이것은 소프트웨어 교육 내용을 알고 있는 교사들은 실과 교과에서 실시하는 것이 타당하지 않다고 생각하는 것을 알 수 있다.

또한 소프트웨어 교육 17시간에 대한 타당도는 컴퓨터 교육을 심화과정을 한 교사들이나 소프트웨어 교육의 내용을 알고 있는 교사들은 17시간으로 할 수 없다고 생각하는 것을 의미한다.

3.4.3 교사들의 지식

소프트웨어 교육 내용을 초등학교 교사들이 얼마나 알고 있는지에 대한 설문 결과는 <Table 6>과 같다.

설문 내용은 다음과 같다.

- 1) 나는 소프트웨어 교육의 내용을 알고 있다.
- 2) 나는 소프트웨어 교육을 하기 위한 평가 방법을 알고 있다.
- 3) 나는 소프트웨어 교육을 위해서 교육 자료를 만들 수 있다.
- 4) 나는 소프트웨어 교육을 위해 교사 연수를 받을 수 있다.

<Table 6> Knowledge of Teacher

| items | not strongly agree | not agree | normal | agree | strongly agree |
|-------|--------------------|-----------|--------|-------|----------------|
| 1) | 23 | 48 | 55 | 49 | 24 |
| 2) | 23 | 62 | 46 | 51 | 0 |
| 3) | 31 | 62 | 47 | 47 | 12 |
| 4) | 25 | 35 | 53 | 62 | 24 |
| 5) | 0 | 10 | 24 | 81 | 84 |

<Table 6>을 분석하여 보면 다음과 같다. 2018년부터 실시할 소프트웨어 교육에 대한 일반적인 지식문제이다. 교사들이 소프트웨어 교육의 내용을 알고 있다고 답한 비율은 37%이고, 잘 모르고 있다고 답한 비율은 34%이다. 소프트웨어 교육을 하기 위한 교수·학습 방법을 알고 있다고 답한 비율은 26%이고, 잘 모르고 있다고 답한 비율은 43%이다. 소프트웨어 교육을 하기 위한 평가 방법을 알고 있다고 답한 비율은 30%이고, 잘 모르고 있다고 답한 비율은 47%이다. 소프트웨어 교육을

위해서 교육 자료를 만들 수 있다고 답한 비율은 43%이고, 잘 모르고 있다고 답한 비율은 30%이다.

소프트웨어 교육을 위해서 소프트웨어 교육 내용, 교수·학습 방법, 평가 방법, 자료 만드는 방법을 잘 모르는 교사들이 많다. 이제 실과 교과에서 소프트웨어 교육을 해야 하기 때문에 모든 교사들은 이러한 내용들을 알아야 한다.

교사들의 83%가 소프트웨어 교육을 위해 교사 연수를 받을 수 있다고 응답했기 때문에 이러한 문제는 교사 연수를 통해서 해결할 수 있을 것이다.

교사들이 경력별로 소프트웨어 교육 내용을 알고 있는지에 대한 t검정 결과는 <Table 7>과 같다. <Table 7>은 교사들의 경력에 따라 소프트웨어 교육 내용을 알고 있는 정도에 차이가 있다는 것을 알 수 있다.

<Table 7> Analysis on recognition of the SW contents

| item | Avg | N | variance | t | Sig |
|-------|------|-----|----------|-------|---------|
| 교사 경력 | 8.24 | 199 | 38.04 | 11.73 | 0.00*** |
| 1) | 3.02 | 199 | 144 | | |
| Total | | 398 | | | |

3.4.4 소프트웨어 교육 내용에 대한 인식

교사들이 2018년부터 실시할 소프트웨어 교육에 대한 내용체계에 대해서 얼마나 많이 인식하는지 아는 것이 중요하다. 이에 대한 설문 분석 결과는 다음 <Table 8>과 같다.

설문 항목은 다음과 같다.

- 1) 일반적인 지식에 기술된 내용이 소프트웨어 교육을 잘 설명하고 있다.
- 2) 내용 요소들을 잘 설명하고 있다.
- 3) 내용 요소인 ‘소프트웨어의 이해’를 보고 가르칠 내용을 알고 있다.
- 4) 내용 요소인 ‘절차적 문제해결’을 보고 가르칠 내용을 알고 있다.
- 5) 내용 요소인 ‘프로그래밍 요소와 구조’를 보고 가르칠 내용을 알고 있다.

<Table 8> Contents of software Education

| items | not strongly agree | not agree | normal | agree | strongly agree |
|-------|--------------------|-----------|--------|-------|----------------|
| 1) | 19 | 71 | 74 | 34 | 1 |
| 2) | 14 | 53 | 82 | 44 | 6 |
| 3) | 26 | 72 | 51 | 43 | 7 |
| 4) | 26 | 67 | 51 | 45 | 10 |
| 5) | 30 | 68 | 50 | 40 | 11 |

<Table 8>의 분석 결과는 다음과 같다. 교육부의 내용체계를 설명하고 교사들에게 질문한 내용이다. “일반적인 지식에 기술된 내용이 소프트웨어 교육을 잘 설명하고 있다”에 대한 답한 비율은 18%이고, 그렇지 않다고 답한 비율은 45%이다.

“내용 요소들을 잘 설명하고 있다.”에 대한 답한 비율은 25%이고, 그렇지 않다고 답한 비율은 34%이다.

“내용 요소인 ‘소프트웨어의 이해’를 보고 가르칠 내용을 알고 있다.”에 대한 답한 비율은 25%이고, 그렇지 않다고 답한 비율은 49%이다.

“내용 요소인 ‘절차적 문제해결’을 보고 가르칠 내용을 알고 있다.”에 대한 답한 비율은 28%이고, 그렇지 않다고 답한 비율은 47%이다.

“내용 요소인 ‘프로그래밍 요소와 구조’를 보고 가르칠 내용을 알고 있다.”에 대한 답한 비율은 26%이고, 그렇지 않다고 답한 비율은 49%이다.

이를 종합해 보면 교육과정에서 정확히 무엇을 가르쳐야 할지 모르는 교사들이 50%에 가깝고 25% 정도만 내용 요소를 알 수 있다는 것이다.

3.4.5 소프트웨어 성취기준에 대한 인식

다음은 교사들이 2018년부터 실시할 소프트웨어 교육에 대한 5개의 성취기준에 대한 설문 조사로서 5개의 성취기준을 이해하는 정도, 교수 학습 자료 개발 정보, 학생들에게 지도할 수 있지에 대한 질문으로 구성되었다.

질문 항목은 다음과 같다.

- 1) “소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.” 성취기준이 이해된다.
- 2) “절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.” 성취기준이 이해된다.

- 3) “프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.” 성취기준이 이해된다.
- 4) “자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.” 성취기준이 이해된다.
- 5) “문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.” 성취기준이 이해된다.
- 6) “소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.” 성취기준에 적합한 교수·학습 자료를 개발할 수 있다.
- 7) “절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.” 성취기준에 적합한 교수·학습 자료를 개발할 수 있다.
- 8) “프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.” 성취기준에 적합한 교수·학습 자료를 개발할 수 있다.
- 9) “자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.” 성취기준에 적합한 교수·학습 자료를 개발할 수 있다.
- 10) “문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.” 성취기준에 적합한 교수·학습 자료를 개발할 수 있다.
- 11) “소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.” 성취기준에 적합하게 학생들을 지도할 수 있다.
- 12) “절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.” 성취기준에 적합하게 학생들을 지도할 수 있다.
- 13) “프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.” 성취기준에 적합하게 학생들을 지도할 수 있다.
- 14) “자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.” 성취기준에 적합하게 학생들을 지도할 수 있다.
- 15) “문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.” 성취기준에 적합하게 학생들을 지도할 수 있다.

내용체계에 대해서 얼마나 많이 인식하는지 아는 것

이 중요하다. 이에 대한 설문 분석 결과는 다음 <Table 9>와 같다.

<Table 9> Achievements of software Education

| item | not strongly agree | not agree | normal | agree | strongly agree |
|------|--------------------|-----------|--------|-------|----------------|
| 1 | 2 | 21 | 48 | 91 | 37 |
| 2 | 14 | 38 | 63 | 58 | 26 |
| 3 | 12 | 20 | 53 | 86 | 28 |
| 4 | 8 | 21 | 50 | 89 | 31 |
| 5 | 11 | 24 | 52 | 77 | 35 |
| 6 | 6 | 28 | 54 | 78 | 33 |
| 7 | 16 | 45 | 74 | 47 | 17 |
| 8 | 20 | 51 | 56 | 51 | 21 |
| 9 | 20 | 49 | 60 | 49 | 21 |
| 10 | 19 | 57 | 56 | 49 | 18 |
| 11 | 3 | 13 | 57 | 88 | 38 |
| 12 | 10 | 32 | 70 | 62 | 25 |
| 13 | 17 | 33 | 57 | 69 | 23 |
| 14 | 15 | 31 | 58 | 71 | 24 |
| 15 | 15 | 30 | 68 | 66 | 20 |

<Table 9>를 분석하여 보면 세 영역으로 나눌 수 있는데, 성취기준을 이해하는 영역과 성취기준을 만족하는 교수·학습 자료를 만드는 영역, 그리고 성취기준대로 지도할 수 있는 능력이다.

성취기준을 이해하는 능력에서 “소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.”는 성취기준이 이해된다고 답한 비율은 64%이다. “절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.”는 성취기준이 이해된다고 답한 비율은 42%이다. “프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.”는 성취기준이 이해된다고 답한 비율은 57%이다. “자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.”는 성취기준이 이해된다고 답한 비율은 60%이다. “문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.”는 성취기준이 이해된다고 답한 비율은 56%이다.

다음은 성취기준을 만족하는 교수·학습 자료를 만드는 영역이다. “소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.”는 성취기준에 적합한 교수·학습 자료를 개발할 수 있다고 답한 비율은

56%이다. “절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.”는 성취기준에 적합한 교수·학습 자료를 개발할 수 있다고 답한 비율은 31%이다. “프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.”는 성취기준에 적합한 교수 학습 자료를 개발할 수 있다고 답한 비율은 36%이다. “자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.”는 성취기준에 적합한 교수·학습 자료를 개발할 수 있다고 답한 비율은 35%이다. “문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.”는 성취기준에 적합한 교수·학습 자료를 개발할 수 있다고 답한 비율은 34%이다.

마지막으로 성취기준대로 지도할 수 있는 능력이다. “소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.”는 성취기준에 적합하게 학생들을 지도할 수 있다고 답한 비율은 63%이다. “절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.”는 성취기준에 적합하게 학생들을 지도할 수 있다.에 대한 답한 비율은 44%이다. “프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.”는 성취기준에 적합하게 학생들을 지도할 수 있다고 답한 비율은 46%이다. “자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.”는 성취기준에 적합하게 학생들을 지도할 수 있다고 답한 비율은 48%이다. “문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.”는 성취기준에 적합하게 학생들을 지도할 수 있다고 답한 비율은 43%이다.

위의 내용을 분석하여 보면 응답자의 50%가 소프트웨어 교육의 성취기준을 이해한다고 하였으나 교수·학습 자료를 개발할 수 있다고 한 비율은 30%에 불과하였으며, 성취기준에 적합하게 지도할 수 있다고 답한 비율도 40%대인 것을 알 수 있다. 따라서 소프트웨어 교육을 잘 하기 위해서 교사들에게 성취기준을 이해하는 연수를 하고, 교수·학습 자료 개발 및 교수·학습 지도 방법 등을 연수하여야 한다.

4. 결론 및 제언

2015년 개정 교육과정 중 소프트웨어 교육과 관련하여 초등학교 교사 199명을 대상으로 인식을 조사한 결과는 다음과 같다.

초등학교 교사들은 72%가 초등학교에서 소프트웨어 교육이 필요하다고 인식하고 있고, 특히 대학에서 심화 과정을 컴퓨터 교육으로 이수한 교사들은 79%가 소프트웨어 교육이 강력히 필요하다고 인식하고 있다. 또한 86%의 교사들이 우리 사회에서 소프트웨어가 중요하다는 것을 인식하고 있다는 것을 알 수 있다. 그런데 2018년부터 초등학교에서 소프트웨어 교육을 실시한다는 것을 모르고 있는 교사들이 29%이고 잘 알고 있는 교사들은 35%이다.

실과 교과에서 소프트웨어 교육을 실시해야 한다고 답한 교사는 50%이고 그렇지 않는 교사가 50% 정도이다. 소프트웨어 교육 내용을 알고 있는 교사들은 37%에 불과하다. 소프트웨어 교육 내용을 알고 있는 교사가 소프트웨어 교육을 실과에서 해야 한다는 것을 t검증 값으로 보면 실과 교과에서 실시하는 것이 타당하지 않다는 것을 알 수 있다.

소프트웨어 교육 시간과 관련해서는 17시간이 충분하지는 않지만 만족하는 교사가 33%이고 부족하다고 생각하는 교사가 41%이므로 17시간으로는 소프트웨어 교육이 부족하다는 것을 알 수 있다.

초등학교 교사들이 소프트웨어 교육에 대한 성취기준을 이해하는 정도는 50%이고, 학생들을 성취기준대로 지도할 수 있는 정도는 40%이고, 교수 학습 자료를 개발할 수 있는 교사는 30%이다.

본 연구의 제언은 다음과 같다.

첫째, 초등학교에서 소프트웨어 교육을 실시하기 위해서 교사의 전문성 신장을 위한 교사 연수가 필수적이다.

둘째, 초등학교에서 소프트웨어 교육이 필수적으로 요구되고 미래사회에 필요하기 때문에 중등과의 연계 등을 다각적으로 고려했을 때 소프트웨어 교육은 17시간 이상 실시하여 독립적인 교과를 만들어야 한다.

참고문헌

[1] Blake Masters & Peter Thiel (2014). Zero to One: Notes on Startups, or How to Build the Future, Crown Business.

[2] CAS (2013). Computing in the national curriculum: A guide for primary teachers. Computing At School.

[3] CAS (2013). Computing in the national curriculum: A guide for secondary teachers. Computing At School.

[4] Deborah Seehorn, Stephen Carey, Daniel Moix, Brian Fuschetto, Irene Lee, Dianne O’Grady-Cuniff, Chris Stephenson, Anita Verno (2011). CSTA K-12 Computer Science Standards Revised 2011.

[5] Hong M.H. (2013). A Study on ICT Competences of Turkey and Korea Focus on PISA 2009 and PISA 2012. *Journal of The Korean Association of Information Education*, 18(1), 151-160.

[6] <http://www.informatikstandards.de/>

[7] Jeong Youngsik, etc (2015). Development of the Software Education Curriculum Model for Elementary Schools. *Journal of The Korean Association of Information Education*, 19(4), 261-270.

[8] KAIE (2014). Information Contents Framework in Elementary Schools.

[9] KERIS (2015). Guidelines of Software Education Management.

[10] Kim Kapsu, etc (2014). A Study on Contents of Information Science Curriculum. *Journal of The Korean Association of Information Education*, 18(1), 161-171.

[11] Kim, Kapsu (2015). A Study on ICT Competences of Korean Students Focus on PISA 2009 and 2012. *Journal of The Korean Association of Information Education*, 19(2), 233-242.

[12] Ministry of Education (2015). The revised national curriculum 2015 for Primary and Secondary Schools, Vol.2015-75, No. 10.

[13] OECD (2013). PISA2012 Results. www.oecd.org/pisa.

[14] 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.

저자소개



김 갑 수

1985 서울대학교계산통계학과(학사)
 1987 서울대학교 계산통계학과 전산학전공(석사)
 1996 서울대학교 계산통계학과 전산학전공(박사)
 1987~1992 삼성전자 사원-과장
 1995~1998 서경대학교 전임강사-조교수
 1998~현재 서울교육대학교 컴퓨터교육과 조교수-교수
 관심분야: 컴퓨터 교육, SW 공학, 정보 영재, 기능성 게임
 e-mail: kskim@snue.ac.kr

