Journal of The Korean Association
of Information Education Vol. 23, No. 3,
June 2019, pp. 245-254

http://dx.doi.org/10.14352/jkaie.2019.23.3.245

예비 교사들의 소프트웨어 교육의 학습 요소 정의 및 효과 분석

김갑수
서울교육대학교 컴퓨터교육과

요 약

2015년 개정 교육과정에서 초등학교 5학년 또는 6학년에서 소프트웨어 교육을 17시간 필수적으로 실시하여야 한다. 따라서 초등 예비교사들이 소프트웨어 교육을 할 수 있어야 한다. 본 연구에서는 초등 예비교사들이 소프트웨어 교육을 할 수 있는 성취기준별로 학습 요소들을 정의한다. 학습 요소별로 예비교사들에게 수업을 실시하였다. 그 결과는 학습 요소별로 교수 학습 및 평가 자료를 만들 수 있었다. 또한 예비교사들이 이들 자료를 기반으로 초등학교에서 소프트웨어 교육을 지도 할 수 있는 능력이 향상되었다는 것을 알 수 있었다.

키워드 : 소프트웨어 교육, 예비교사, 초등학생, 교수 학습 자료, 평가 자료

An Analysis and Definition of Software Education Learning Elements for Pre_service Elementary Teacher

Kapsu Kim
Dept. of Computer Education, Seoul National University of Education

ABSTRACT

In the 2015 revision curriculum, software education must be conducted for 17 hours in elementary school fifth or sixth grade. Pre_service elementary teachers should be able to teach software. In this study, learning factors according to the achievement criteria for software education will be defined for pre-service elementary teachers. Pre-service teachers were taught by learning elements. As a result, the pre_service teachers were able to create teaching and evaluation materials for each learning element. It was also found that the pre-service teachers improved their ability to teach software education in elementary school based on these materials.

Keywords : Software Education,Pre_service Teacher, Elementary Student, Teaching Materials, Evaluation Materials

논문투고 : 2019-06-18
논문심사 : 2019-06-24
심사완료 : 2019-06-24

1. 서론

21세기 지식 정보 사회, 지능정보 사회, 4차 산업 혁명에서 교육에 대한 중요성DMFH 매우 많은 것이 언급되고 있는 것들의 공통적인 요소가 소프트웨어의 중요성이다.

이에 따라 미국의 경우에는 CTSA와 NSTA에서 컴퓨터 교육 과정을 유치원부터 실시하는 것을 권고하고 있다. ACM의 CSTA에서 초등학교부터 고등학교까지 어떤 내용으로 가르치고 있는지에 대한 성취 기준을 제안하였다. 2011년에 제안한 초등학교 컴퓨팅 교육과정을 만들었다[4,5,16]. 이를 수정하여 2016년도에 표준안을 발표하였고, 2017년도에 최종적으로 새로운 표준을 만들어 유치원부터 컴퓨터 교육을 실시하는 것을 권고하고 있다.

영국의 경우에는 1992년 국가 교육과정을 설정하면서 정보통신기술을 필수교과로 규정하여 체계적으로 정보통신 기술의 활용 교육위주로 진행하여 왔다. 그 이후에 2014년부터 정보통신기술 교육에서 컴퓨팅 교육이라는 것으로 이름을 변경하여 실제 프로그램 개발위주의 교육을 초등학교 1학년부터 체계적으로 실시하고 있다[1,2,3].

우리나라는 2000년에는 새로운 지식 정보 사회를 위해서 학생들이 정보 통신 기술 교육을 잘 교육하기 위하여 교육부에서는 정보통신운영지침[7,8]을 만들었다. 이 때에는 7차 교육과정이 완성된 단계에서 정부의 강력한 지침으로 초등학교 1학년부터 6학년까지 모든 학생들이 1시간이상 교육받을 수 있게 만들어 교육과정 이외에 초등학교에서 수업을 할 수 있게 만들었다. 이를 개정하여 2005년에는 좀 더 능동적으로 정보처리를 하는 능력을 추가하여 학생들이 정보통신교육을 받을 수 있게 하였지만 정식 교육과정에서 실시하지 않은 단점이 있었다. 이에 2008년 정부의 규제 개혁차원이 지침을 폐지하여 컴퓨터 교육이 초등학교 현장에서는 실종되었다.

이런 상황에서 2017년에 OECD에서 발표한 PISA ICT 배경통계[11,12,13,14,15]를 보면 우리나라 학생들의 ICT 기반이 매우 미약하다. 학교에서 ICT 접근성은 25.48%이고 집에서 ICT 접근성은 49.64%이다. 이 통계는 OECD 30개 국가 중에 28위이다. 학교에서 ICT 활용성은 2.718%이고, 학교 이외에서 ICT 활용성은 14.00%으로 OECD 31개 국가 중에 31위이다. 이런 상황에서 ICT에 대한 국가적인 교육 틀이 필요하고 예비교사들이 초중등 교육에서 이미 다른 교과와 다르게 인지도가 매우 적다는 것은 시사하는 점이 매우 크다고 할 수 있다.

이에 우리나라는 2015년 개정 교육과정[14,15]을 만들면서 초등학교에서 17시간의 소프트웨어 교육을 실시하게 하였고, 2019년도는 초등학교 6학년에 소프트웨어 교육을 실시하고 있다. 이에 예비교사를 양성하는 교육대학에서 소프트웨어 교육을 할 수 있는 교사들을 양성해야 하는 것이 필수적이다.

소프트웨어 교육을 잘 하기 위해서 예비교사들의 소프트웨어 교육 능력이 매우 중요하다. 예비교사들의 소프트웨어 교육 능력을 기르기 위해서 2015년 개정 교육과정에서 실과 교과의 소프트웨어 교육 부분의 5개의 성취기준에 대해 예비교사들이 100%이해하고 예비교사들이 성취 기준별로 교수 학습 자료를 만들 수 있어야 하고, 평가 자료를 잘 만들 수 있어야 하고 이를 기반으로 초등학생들에게 소프트웨어 교육할 수 있는 능력을 기르게 해야 한다.

따라서 본 연구에서는 성취기준별로 학습 요소들을 추출하여 2학점 수업을 5개의 성취기준별로 학습 요소별로 수업을 한 후에 그 효과는 예비교사들의 이해도, 교수학습 자료 개발 능력, 평가 자료 개발 능력, 현장에서 초등학생들이 지도할 수 있는 능력을 t-검증으로 평가한다.

제2장에서는 2015개정 교육과정의 성취기준 5개와 그에 대한 해설서를 설명한다, 제3장에서는 성취기준별로 학습 요소들을 추출하였고, 각 요소별로 교수 학습 자료를 만들어 학생들에게 수업한 결과의 효과 분석을 한다. 제4장에서는 본 연구의 결론을 맺는다.

2. 관련 연구

2.1 개요

우리나라 컴퓨터 교육과정은 한국정보교육학회 등에서 초등학교 1학년부터 실시하고, 독립교과로 구성하자는 주장이 많았지만[6], 2015 개정교육과정에서는 초등학교 5학년부터 6학년까지 컴퓨터 교육을 실시하도록 되어 있고[9,10], 각 내용은 2019년부터 실질적으로 소프

트웨어 교육을 실시하게 되어 있다. 교육부 교육과정의 기본 틀에는 소프트웨어 교육이 기술 시스템 영역으로 되어 있고, 일반화된 지식으로는 '통신 기술은 정보를 생산, 가공하여 다양한 수단과 장치를 통하여 송수신하여 공유한다.'로 되어 있고, 내용 체계는 소프트웨어 이해, 절차적 문제 해결, 프로그래밍 요소와 구조로 되어 있다. 이는 아무리 보아도 소프트웨어 개발 또는 코딩 교육과정에 맞추어져 있지 소프트웨어를 담고 있는 컴퓨터 시스템 교육과정은 찾아 볼 수 없다는 단점이 있다.

2015 개정교육과정의 구체적인 성취 기준은 다음과 같다[9,10]. 다음 각 절에 개정 교육과정을 그대로 기술한 것이다.

2.2 성취기준1

성취기준1은 "소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다."로 되어 있고, 그 해설은 "컴퓨터에 사용된 소프트웨어 이외에도 휴대폰, 가전제품, 사물인터넷 제품까지 여러 상황에서 사용되는 소프트웨어를 탐색해 보고 우리 생활에 미치는 영향을 이해한다."로 되어 있다[9.10]. 이 성취기준을 지도할 수 있는 예비 교사들의 수업 내용과 교수 학습 방법 및 평가 방법으로 을 2주차로 구성하여 실시하였다.

2.3 성취기준2

성취기준2는 "절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다."로 되어 있고, 그 해설은 "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."로 되어 있다[9,10]. 본 연구에서는 이 성취 기준을 지도할 수 있는 예비 교사들의 수업 내용을 2주차로 구성하여 실시하였다.

2.4 성취기준3

성취기준2는 "프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다."로 되어 있고, 그 해설은 "블록 기반의 교육용 프로그래밍 도구를 활용하여 기초적인 프로그래밍 과정을 체험 하고 자신만의 간단한 프로그램을 만들어 본다."로 되어 있다[9,10]. 본 연구에서는 이 성취 기준을 지도할 수 있는 예비 교사들의 수업 내용을 2주차로 구성하여 실시하였다.

2.5 성취기준4

성취기준4는 "자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다."로 되어 있고, 그 해설은 "수치 값을 입력하여 덧셈이나 뺄셈의 결과를 출력하거나, 복수의 문자열을 입력 하여 두 문자열을 서로 연결한 결과를 출력하는 프로그램을 만들어 봄으로써, 소프트웨어의 입력, 처리, 출력 과정을 이해한다."로 되어 있다[9,10]. 본 연구에서는 이 성취 기준을 지도할 수 있는 예비 교사들의 수업 내용을 3주차로 구성하여 실시하였다.

2.6 성취기준5

성취기준2는 "문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다."로 되어 있고, 그 해설은 "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."로 되어 있다[9,10].

3. 연구 내용 및 결과 분석

3.1 연구 대상

본 연구에서는 S교육대학교 4학년 학생들의 3반에 대한 한 학기 수업을 소프트웨어 성취기준별로 학생들이 학습 요소들을 정의하고 이를 수업한 후에 사전 테스트와 사후 테스트를 한 후에 t-검증을 실시하였다. 연구 대상의 구체적인 데이터는 3반 학생들이 수업 전에 74명이 설문에 참석하였고, 수업 후에는 한명이 참석하지 않아 73명이 참석하여 총 147명이 참석하였다.

<Table 1> Number of Classes

	Before	After	Total
A class	23	29	52
B class	24	20	44
C class	27	24	51
Total	74	73	147

3.2 수업 내용 구성

예비교사들이 2학점으로 총 15주 수업을 구성하였고, 주당 2시간 수업을 구성하였다.

5개의 성취기준별로 예비교사들이 이해할 수 있고, 교수학습자료를 개발할 수 있고, 교육할 수 있고, 평가자료를 만들 수 있어야 하고, 성취도가 높은 학생들과 그렇지 않은 학생들을 지도할 수 있어야 하고, 협동학습과 프로젝트기반 학습으로 지도할 수 있어야 한다. 또한 각 성취기준별로 예제를 들어 지도할 수 있어야 한다.

1주차 및 2주차 수업은 성취기준1에 대한 강의 내용과 교수학습 방법 및 평가 방법으로 구성하였다. 구체적인 내용은 다음과 같다.

성취기준에 대한 학습 요소는 다음과 같다.

1) 소프트웨어와 하드웨어가 무엇인지 이해할 수 있다.
2) 소프트웨어가 없으면 어떤 변화가 있을 것인지 이해할 수 있다.
3) 가정, 학교, 사회에서 사용하는 기기들 중에서 소프트웨어가 적용된 것과 아닌 것을 구별할 수 있다.
4) 소프트웨어가 있음으로 어떤 점이 편리한지를 이해하고 설명할 수 있다.
5) 소프트웨어의 종류들을 분류할 수 있다.

위의 학습 요소별로 예비 교사들이 필요한 교육 자료를 만드는 방법과 평가 문항들을 만드는 방법을 교육하고, 교수 학습 방법으로 협동학습 또는 프로젝트 학습을 할 수 있는 것을 교육한다.

프로젝트 학습 또는 협동학습 모형의 주제들은 다음과 같은 것을 교육한다.

1) 소프트웨어가 직업의 변화를 어떻게 시키고 있는지 알아본다.(교사, 의사, 택시기사, 야구 선수, 예능 전문가, 사무원, 택배 기사)
2) 소프트웨어가 일을 변화시키는 것을 조사한다.(가정, 학교, 공장, 야구장 등)
3) 각종 기기에는 어떤 소프트웨어들이 있는가 조사한다.(TV, 냉장고, 자동차, 핸드폰, 엘리베이트, 신호등)
4) 각 교과 수업에 소프트웨어를 사용하면 좋은 점을 조사하고, 실제 수업에 사용한 후에 느낀점을 이야기 한다.(과학 등 각종 실험 데이터를 소프트웨어를 이용하여 정리하기)

3주 및 4주차 수업은 성취기준2에 대한 강의 내용과 교수학습 방법 및 평가 방법으로 구성하였다. 구체적인 내용은 다음과 같다.

먼저 성취기준에 대한 학습 요소는 다음과 같다.

1) 절차적인 사고와 절차적인 사고를 하지 않을 때에 어떤 일이 일어 났는지를 안다.
2) 생각의 단위 즉 처리 단위를 정할 수 있다.
3) 절차적인 사고를 표현하는 방법을 안다.
4) 프로차트를 작성하는 방법을 한다.
5) 가상 코드가 무엇인지 알고 표현하는 방법을 안다.
6) 프로차트와 가상코드를 서로 변환하는 할 수 있다.

이 때 프로차트<흐름도>를 작성할 때에 다양한 도구를 사용할 수 있게 안내한다. 흐름도를 작성할 때에 복잡한 기능들보다 가장 기본적인 기호인 시작 또는 종료 기호, 처리 기호, 입력 또는 출력 기호, 판단 기호인 4개의 기호만 학습하게 한다.

다음과 같은 수업 사례로 프로젝트 학습 또는 협동학습 모형의 사례는 다음과 같은 주제이다..

“나이가 20세 이하이면 3000원 입장료를 20% 할인해 준다.”과정을 절차적인 사고를 하는 것을 연습한다.

첫 번째 사고의 단위(명령어의 단위)를 찾아낸다.

- 나이를 입력 받는다.
- 티켓값을 3000원으로 정한다.
- 나이가 20보다 작은 지 판단한다.
- 나이가 20보다 작으면 티켓값을 2400원으로 정한다.
- 티켓값을 출력한다.

위의 5개의 명령어들의 집합을 생각하게 한다.

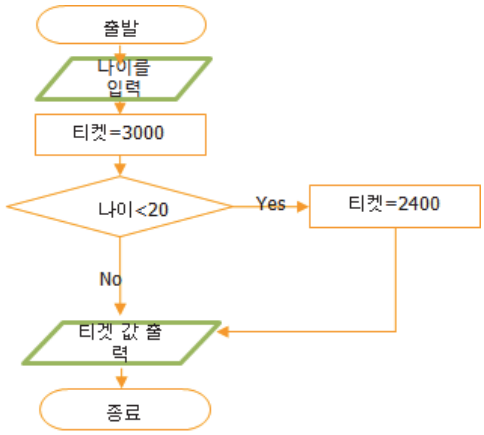
위의 5개의 명령어들의 순서를 변경하면 어떻게 되는지? 순서를 변경해도 되는지 등을 생각하게 한다.

위의 명령어를 기반으로 가상코드들 작성하게 하고

그 결과는 다음과 같다.

나이를 입력받는다
티켓값을 3000원으로 정한다.
나이가 20보다 작은지 판단한다
참이면 티켓값을 20% 할인한다.
티켓값을 출력한다.

이 가상 코드를 흐름도를 작성하면 다음 <Figure 1>과 같다. 학생들이 가상 코드와 흐름도를 서로 같이 표현 할 수 있어야 한다.



(Fig. 1) Basic Flowchart

5주 및 6주차 수업은 성취기준3에 대한 강의 내용과 교수학습 방법 및 평가 방법으로 구성하였다. 구체적인 내용은 다음과 같다.

먼저 성취기준에 대한 학습 요소는 다음과 같다.

1) 프로그래밍 도구인 개발환경에 대해 이해한다.
2) 생산성 향상도구인 워드프로세싱과 차이점이 무엇인지 이해한다.
3) 파일을 만들고 저장하고 실행하는 것을 할 수 있다.
4) 기초적인 프로그램을 선생님이 하는 것을 따라하기를 할 수 있다.
5) 기초적인 프로그램을 수정할 수 있다.
6) 기초적인 프로그램의 각 기능이 무엇인지 알 수 있고, 또한 설명할 수 있다.
7) 기초적인 프로그램을 흐름도로 바꿀 수 있다.
8) 기초적인 프로그램을 가상코드로 작성할 수 있다.

기초적인 프로그램에서 학생들이 재미있어 하는 게임의 필수 요소인 방향키에 따라 야구공 등이 움직이게 하는 방법을 처음 접하는 문제이다.

이 문제의 프로그램 코드는 다음 <Figure2>와 같다.

<Figure2>의 프로그램을 유사코드 변경하게 하면 다음과 같다.

(Fig. 2) Program Code

1단계: Octopus 를 (0,0)에 둔다
2단계: 무한 반복한다.
2.1단계: 왼쪽 방향키를 누르면 x값이 10감소한다.
2.2단계: 오른쪽 방향키를 누르면 x값이 10증가한다.
2.3단계: 위쪽 방향키를 누르면 y값이 10증가한다.
2.4단계: 아래쪽 방향키를 누르면 y값이 10감소한다.

<Figure 2>의 프로그램을 학생들이 자신의 능력에 따라 변경하게 하고, 이를 기반으로 학생들에게 토론하게 한다.

7주부터 9주차 수업은 성취기준4에 대한 강의 내용과 교수학습 방법 및 평가 방법으로 구성하였다. 구체적인 내용은 다음과 같다.

먼저 성취기준에 대한 학습 요소는 다음과 같다.

1) 컴퓨터가 자료를 처리하는 과정을 이해한다.

2) 데이터를 화면으로부터 입력 받는 것을 알 수 있고, 또한 화면으로부터 데이터를 입력받을 수 있다.

3) 변수의 개념을 이해한다.

4) 변수의 이름을 붙일 수 있다.

5) 변수에 데이터를 저장하는 방법을 알 수 있다.

6) 변수 값을 사용할 수 있다.

7) 산술 연산자를 이해하고 사용할 수 있다.

8) 수식을 세울 수 있다.

9) 수식의 결과값을 저장할 수 있다.

10) 데이터를 화면에 출력하게 할 수 있다.

이에 대한 간단한 예제는 국어와 수학 점수를 입력받아 평균을 계산하여 평균을 출력하는 프로그램을 만들어 보자.

이 활동에서는 변수를 찾는 연습을 한다. 여기서 변수는 국어, 영어, 평균의 세 개의 변수가 있을 수 있다.

다음은 연산자를 이용하여 수식을 세워야 한다.

평균= (국어+ 수학)/2

다음과 같은 명령어 단위를 생각하여 가상 코드를 만들 수 있을 것이다.

~를 클릭한다.
데이터 블럭
국어 변수를 0을 만든다.
수학 변수를 0을 만든다.
평균 변수를 0을 만든다.
국어 점수는? 묻고 기다린다.
국어 변수에 대답을 설정한다.
수학 점수는? 묻고 기다린다.
수학 변수에 대답을 설정한다.
평균은 아마 아마... 10초 정도 생각한다.
평균= (국어+수학)/2를 설정한다.
평균은 아마 얼마일 것이다.

위의 가상 코드를 이용하여 스크래치 코드는 다음 <Figure 3>과 같다.

(Fig.e 3>)Mean Program

위의 가상 코드를 이용하여 파이썬 코드도 만들어 보게 하고 실습하게 한다.

```
kor=0
math=0
average=0
kor=int(input("국어 점수를 입력하시오"))
math=int(input("수학 점수를 입력하시오"))
print("평균은 아마\n")
average=(kor+math)/2
print(average)
```

각 언어로 구현한 것을 서로 이야기 하게 해 보고 예비교사들이 블록코딩뿐만 아니라 텍스트 기반 언어도 이해하게 한다.

10주부터 14주차 수업은 성취기준5에 대한 강의 내용과 교수학습 방법 및 평가 방법으로 구성하였다. 구체적인 내용은 다음과 같다.

먼저 성취기준에 대한 학습 요소는 다음과 같다.

1) 순차적으로 명령어들을 처리하는 것을 이해할 수 있다.
2) 관계 연산자들을 이해하고, 사용할 수 있다.
3) 논리연산자들을 이해하고, 사용할 수 있다.
3) 반복적으로 명령어들을 처리하는 것을 이해할수 있고 사용할 수 있다.
4) 선택적으로 명령어들을 처리하는 것을 이해할 수 있고, 사용할 수 있다.
5) 참일 때만 수행하는 것을 이해할 수 있고, 프로그래밍 할 수 있다.
6) 거짓일 때만 수행하는 것을 이해할 수 있고, 프로그래밍 할 수 있다.
7) 참과 거짓일 때에 다른 것을 수행하는 것을 이해할 수 있고, 프로그래밍 할 수 있다.
8) 순차적으로 사용하는 것과 반복적으로 사용하는 것의 장점과 단점을 이해 할 수 있다.
9) 반복적인 것을 찾아서 프로그래밍할 수 있다.
10) 같은 명령어를 10번 또는 특정 횟수만큼 반복할 수 있다.
11) 어떤 조건을 만족할 때까지 반복할 수 있다.
12) 반복문 안에서 조건문을 사용할 수 있다.
13) 조건문 안에서 반복문을 사용할 수 있다.
14) 반복문 안에서 반복문을 사용할 수 있다.

이 성취기준에 대한 학습 요소들은 위의 14개를 만들었고, 이 학습 요소들을 학습하기 위한 다양한 문제를 만들어서 예비 교사들의 소프트웨어 수업 능력을 향상시키는 것이 목적이다.

다음과 같은 예제를 기반으로 위의 학습 요소들을 학습할 수 있데 한다.

10개의 한자리수 덧셈 문제를 평가하는 프로그램을 만들어 보는 문제로 한 문제당 10점이 배정되어 점수를 출력하는 문제이다.

이 문제는 반복문 속에서 조건문을 하는 문제라는 것을 알게 한다.

변수는 숫자1 변수, 숫자2 변수, 답 변수, 및 점수 변수 세개가 있다는 것을 알 게 한다.

관계 연산식을 만들어 수 있어야 한다. 이에 대한 가상 코드는 다음과 같다.

어떤 것을 클릭하면 동작하게 한다.[제어블럭]"지금부터는 임의의 한자리 수를 덧셈을 평가는 것입니다."를 스프라이트가 2초 이야기 한다.
점수 변수에 0을 지정한다.
10번 반복하기
숫자1 변수에 임의의 수가 저장된다
숫자2 변수에 임의의 수가 저장된다
스프라이트가 "숫자1 변수 값"과 "숫자2 변수 값 " 을 더하기 하게 하고 답을 입력할 때까지 기다린다.
답 변수에 입력한 값을 저장된다.
숫자1과 숫자2 변수 값의 합이 답 변수 값과 같은지를 체크하는 관계연산자식을 만든다.
식이
참이면 점수 변수에 10점을 증가한다
거짓이면 점수 변수값은 그대로
잠수 변수 값을 출력한다.

이 가상 코드를 스크래치 언어로 표현하면 다음 <Figure 4>와 같다.

(Fig. 4) Example of Criteria 5

3.3 수업 결과 분석

수업 결과 분석으로 성취기준에 대한 이해도, 교수학습 자료를 만드는 능력, 평가 자료를 만드는 능력, 학생지도 능력에 대한 5개의 성취기준별로 수업 전후의

효과를 분석한다. 분석 방법으로 t-검증을 한다.

"성취기준 이해도"에 대한 예비교사들의 분석 결과는 기본 통계는 < Table 2>와 같다.

<Table 2> Understanding the criteria(Basic statistics)

Criteria	group	Number	mean	Standard Deviation	S.E.of Mean
1	Before	74	4.16	.759	.088
	After	73	4.47	.647	.076
2	Before	74	4.12	.827	.096
	After	73	4.51	.669	.078
3	Before	74	4.16	.828	.096
	After	73	4.48	.648	.076
4	Before	74	4.11	.837	.097
	After	73	4.47	.668	.078
5	Before	74	4.15	.839	.098
	After	73	4.48	.689	.081

t-검증 결과는 <Table 3>과 같다.

<Table 3> Understanding the criteria(t-test)

Criteria	t	d.f	Significant probability (both)	Mean difference
1	-2.608	145	.010	-.304
2	-3.102	145	.002	-.385
3	-2.585	145	.011	-.317
4	-2.860	145	.005	-.358
5	-2.610	145	.010	-.331

본 연구에서 학생들이 성취기준에 대한 이해도는 성취기준4와 성취기준2는 유의미한 차이가 있다는 것을 알 수 있다. 전체적으로 학생들이 이미 4학점을 들었기 때문에 수업전에 성취기준 이해도는 4,11에서 4.16으로 매우 높았다. 수업후에 학생들이 성취기준 이해도는 4.47에서 4.51로 높은 편이다. 성취기준2와 성취기준4는 5% 유의수준으로 학습의 효과가 있다는 것을 알 수 있다.

다음은 "교수학습 자료를 만들 수 있는 능력"에 대한 기본 통계 자료는 다음 <Table 4>와 같고, t-검증 결과는 <Table 5>와 같다.

<Table 4> Developing Teaching Material (Basic statistics)

Criteria	group	Number	mean	Standard Deviation	S.E.of Mean
1	Before	74	3.61	.919	.107
	After	73	4.29	.754	.088
2	Before	74	3.54	.863	.100
	After	73	4.27	.750	.088
3	Before	74	3.50	.940	.109
	After	73	4.30	.776	.091
4	Before	74	3.53	.954	.111
	After	73	4.26	.817	.096
5	Before	74	3.59	.920	.107
	After	73	4.32	.743	.087

<Table 5> Developing Teaching Material(t-test)

Criteria	t-value	d.f	Significant probability (both)	Mean difference
1	-4.897	145	.000	-.680
2	-5.496	145	.000	-.733
3	-5.632	145	.000	-.801
4	-5.002	145	.000	-.733
5	-5.218	145	.000	-.720

위의 <Table 4>와 <Table 5>를 분석하여 보면 예비학생들이 수업 전후에 평균의 차이가 모든 성취기준에 유의 수준 5% 이하에 유의미한 효과가 있다는 것을 알 수 있고, 예비교사들의 교수 학습자료 개발 능력이 향상되었다고 볼 수 있다.

다음은 "평가 자료를 만들 수 있는 능력."에 대한 기본 통계 자료는 <Table 6>와 같고, *t*-검증 결과는 <Table 7>와 같다.

<Table 6> Evaluation Material (Basic statistics)

Criteria	group	Number	mean	Standard Deviation	S.E.of Mean
1	Before	74	3.69	.905	.105
	After	73	4.27	.712	.083
2	Before	74	3.61	.873	.101
	After	73	4.25	.741	.087
3	Before	74	3.58	.936	.109
	After	73	4.22	.768	.090
4	Before	74	3.58	.936	.109
	After	73	4.16	.764	.089
5	Before	74	3.59	.920	.107
	After	73	4.32	.743	.087

<Table 7> Evaluation Material(t-test)

Criteria	t	d,f.	Significant probability (both)	Mean difference
1	-4.348	145	.000	-.585
2	-4.777	145	.000	-.638
3	-4.513	145	.000	-.638
4	-4.135	145	.000	-.583
5	-4.702	145	.000	-.638

<Table 6>과 <Table 7>을 살펴보면 예비 교사들이 수업 전후에 평균의 차이가 모든 성취기준에 많이 나고, 모든 성취 기준에 대한 유의 확률이 0이기 때문에 예비 교사들이 소프트웨어 교육에 대한 평가 자료를 만드는데 매우 유의미하다는 것을 알 수 있다.

다음은 "예비 교사들이 학생들을 지도할 수 능력"에 대한 기본 통계 자료는 <Table 8>과 같고, t-검증 결과는 <Table 9>와 같다.

<Table 8> Teaching (Basic statistics)

Criteria	group	Number	mean	Standard Deviation	S.E.of Mean
1	Before	74	3.86	.799	.093
	After	73	4.36	.632	.074
2	Before	74	3.80	.844	.098
	After	73	4.34	.650	.076
3	Before	74	3.73	.865	.101
	After	73	4.29	.656	.077
4	Before	74	3.73	.926	.108
	After	73	4.21	.763	.089
5	Before	74	3.66	.911	.106
	After	73	4.22	.712	.083

<Table 9> Teaching(t-test)

Criteria	t	d,f	Significant probability (both)	Mean difference
1	-4.130	145	.000	-.491
2	-4.384	145	.000	-.545
3	-4.403	145	.000	-.558
4	-3.397	145	.001	-.476
5	-4.127	145	.000	-.557

<Table 8>과 <Table 9>를 분석하여 보면 학생들이 수업 전후에 평균의 차이가 모든 성취기준에 많이 나고, 모든 성취 기준에 대한 유의 확률이 0이기 때문에 예비 교사들이 학생들을 지도 할 수 있다는 것을 알 수 있다.

4. 결론

본 연구에서는 예비교사들이 초등학교에 소프트웨어 교육을 할 수 있는 능력을 기르기 위해서 2015년 개정 교육과정에서 소프트웨어 교육을 할 수 있는 능력을 기르는 것이 필요하다.

예비교사들이 초등학교 현장에서 소프트웨어 교육을 할 수 있는 각 성취기준별로 학습 요소들을 추출하여 예비교사들에게 교육하여 그 효과를 분석하였다.

성취기준1부터 성취기준5까지를 학습하기 위한 학습 요소들을 분석하여 각 요소별로 예비 교사들을 지도하였다.

분석결과로는 예비 교사들의 성취기준 이해도는 이미 컴퓨터 관련 교과목 4학점을 이수하였기 때문에 매우 높았다. 따라서 본 수업 활동으로 유의미 한 차이가 나는 것은 성취기준 2의 절차적인 사고와 성취기준4인 간단한 프로그램을 만들어 보는 것에 대한 유의수준 5%에 대한 의미 있는 결과가 나왔다. 절차적인 사고는 학생들이 처음 접하는 것이고, 간단한 프로그램에 대한 것은 의미가 명확하지 않고 변수, 연산자, 수식을 학습하여야 하는데 이 부분이 성취기준에 대한 표현이 되어 있지 않는 것 같아서 체감적인 느낌인 것 같다.

성취기준별로 학습 요소를 분석하여 예비교사들에게 지도한 결과 예비교사들의 교수학습 자료를 잘 만들 수 있고, 또한 평가 자료도 잘 만들고, 이를 기반으로 예비교사들이 앞으로 학교 현장에서도 수업을 잘 지도 할 수 있다는 것을 모든 항목에서 유의수준 5%로 의미 있는 결과가 나왔다.

참고문헌

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

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

[3] CAS(2012). Computer science : A curriculum for schools, Computing At School.

[4] Deborah Seehorn, Stephen Carey, Daniel Moix,Brian Fuschetto, Irene Lee, Dianne O' Grady-Cuniff, Chris Stephenson, Anita Verno (2016). CSTA K-12 COMPUTER SCIENCE STANDARDS REVISED 2016 CSTA STANDARDS TASK FORCE.

[5] 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.

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

[7] MOE(2000), Manual of ICT in elementary and secondary schools.

[8] MOE(2005), Manual of ICT in elementary and secondary schools.

[9] MOE(2015), 2015 Revised National Curriculum 2015-74 [10]

[10] MOE(2015), 2015 Revised National Curriculum 2015-74(Attached issue 10)

[11] OECD(2003). Feasibility study for the PISA ICT literacy assessment: report to network A. Paris: OECD.

[12] OECD(2009). PISA data analysis manual. Paris: OECD.

[13] OECD(2011). PISA 2009 Results: Students On Line Digital Technologies and Performance (Volume IV).

[14] OECD(2013). PISA2012 Results

[15] OECD(2017). PISA2015 Results

[16] 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

1985.2 서울대학교 계산통계학과 (학사)

1987.2 서울대학교 계산통계학과 전산학전공(석사)

1996.2 서울대학교 계산통계학과 전산학전공(박사)

1987.-1992. 삼성전자 사원-과장

1995.-1998. 서경대학교 전임강사-조교수

1998.-현재 서울교육대학교 컴퓨터교육과 조교수-교수

관심분야 : 컴퓨터 교육, 소프트웨어 공학, 정보 영재, 기능성 게임, 인공지능 교육

E-mail : kskim@snue.ac.kr