

## 예비교사에 적용한 초등정보교과교육 사례 : 프로그래밍 교육 중심으로

한규정

공주교육대학교 컴퓨터교육과

### 요약

본 연구는 예비교사를 양성하는 교육대학교의 ‘초등컴퓨터교육론’ 교과 교육 내용 중 프로그래밍 교육 사례이다. 수업은 온라인으로 진행되었다. 예비교사에게 적용한 프로그래밍 교육방법은 시연중심으로 진행하였으며 예비교사들은 프로젝트 과제로 교과와 융합한 소프트웨어 교수학습 지도안을 제작하였다. 또한 이를 마이크로 티칭으로 발표하였고 평가로서 학생들의 프로그래밍 역량을 측정하기 위하여 온라인 실기 평가를 실시하였다. 예비교사들은 프로그래밍과 수업의 경험을 성찰노트로 작성하여 자신의 약점과 개선할 사항을 점검하였다. 연구 결과로 일부 예비교사들은 프로그래밍에 대해 어려움을 겪고 있었는데 그 내용은 무한반복의 개념 이해, 변수의 사용, 새로운 응용 프로그램의 제작 등으로 조사되었다.

키워드 : 초등정보교육, 코딩, 프로그래밍, 프로그래밍 교육, 엔트리

## A Case Study of Elementary School Informatics Education for Pre-service Teachers : Focusing on Programming Education

Kyujung Han

Dept. of Computer Education, Gong Ju National University of Education

### Abstract

This study is a case study of programming education among the contents of ‘Elementary Computer Education Theory’ of the University of Education, which trains for pre-service teachers. The classes was conducted online. In this study, the programming education method applied to pre-service teacher was Demonstrated based Education, they produced a lesson plan that was fused with the use of software as a project task and presented it through micro-teaching. In addition, an online practical evaluation was conducted to measure pre-teachers’ programming competencies as an evaluation. They wrote their experiences of programming and class as reflection notes to check their weaknesses and areas to improve. As a result of the research, some pre-teachers were having difficulty with programming have been investigated by understanding the concept of infinite repetition, using variables, and creating new application programs.

Keywords : Information Education for Pre-service Teachers, Coding, Programming, Programming Education, Entry

교신저자 : 한규정(공주교육대학교 컴퓨터교육과)

논문투고 : 2021-02-09

논문심사 : 2021-02-09

심사완료 : 2021-02-14

## 1. 연구의 필요성과 목적

세계 각국은 빅 데이터, 인공지능, 자율주행차 등으로 빠르게 발전하고 있는 4차 산업 혁명 사회와 미래를 대비하기 위해 우수 인재 양성 프로그램의 하나로 정보 교육을 강화하고 있다. 특히 소프트웨어 교육은 컴퓨팅 사고력, 협력적 문제해결력, 정보문화 소양 등 과학기술 창조력을 길러주는데 근본이 되며 이는 논리적 사고력 향상을 통한 창의 융합형 인재 양성에 중요하다[12]. 우리나라 정보교육과정으로 소프트웨어교육은 복잡하고 다양한 문제를 해결하기 위한 컴퓨팅 사고를 기반으로 언플러그드, 코딩, 피지컬 컴퓨팅, 그리고 인공지능 교육 등으로 구성되고 있다.

초등 예비교사를 양성하는 교육대학교에서의 정보교육은 교양, 교과, 실기 교육과정으로 실시하고 있으며 특히 소프트웨어 교육에 관련된 교과 학점과 시수 비중을 늘리고 있으나 타교과인 수학, 과학에 비하면 많이 부족한 이수학점을 요구하고 있어 향후 소프트웨어 교육과정의 개선이 요구된다[9].

소프트웨어 교육 중 프로그래밍 교육은 창의적 사고를 통해 문제해결로 이루어지는데 이런 과정 속에 문제 분석, 문제분해, 패턴 인식, 절차적사고, 논리적 사고와 사람들과의 의사소통 능력 등이 향상된다[4]. 그러나 본 연구의 A 교육대학교에서의 전교생 대상 프로그래밍 교육은 이수학점이 2학점내외로 매우 적게 할당되어 있다. 따라서 다양한 프로그래밍 언어 활용 교육에 제약이 되며 소프트웨어와 교과 융합 활동 등의 충분한 시간을 보장되지 않고 있는 실정이다. 특히 프로그래밍 관련 시수부족은 최근 피지컬 컴퓨팅 프로그래밍과 인공지능 프로그래밍 등의 새로운 개념을 짧은 시간만 소개하는 정도로 다루어야 하는 문제점이 발생되고 있다.

본 연구의 목적은 초등 예비교사의 대부분이 코딩 초보자인 점을 감안하여 적은 시수임에도 불구하고 효율적인 프로그래밍 교육 방법을 탐색하는 데 있다. 연구의 접근 방법은 다음과 같다.

첫째, 프로그래밍 교육에 초보자들도 쉽게 접근하는 방법으로 시연중심방법 (Demonstrated Based Education)을 적용한다.

둘째, 프로그래밍과 소프트웨어를 교과에 실제 활용하기 위한 프로젝트로 소프트웨어 교과 융합 교수학습

지도안 작성 프로젝트 과제를 수행한다.

셋째, 소프트웨어교과융합 교수 학습지도안을 마이크로 티칭으로 발표, 토론, 지도조언 등을 실시함으로 실제 적용가능성, 문제점, 개선점 등을 검토한다.

넷째, 프로그래밍과 소프트웨어융합 프로젝트, 마이크로 티칭의 경험을 통해 자신의 약점과 개선할 사항을 발견하는 성찰 노트를 작성한다.

다섯째, 프로그래밍 역량 평가를 위해 부정행위를 방지할 수 있는 공정한 온라인 실기 평가를 실시한다.

여섯째, 예비교사들의 성찰노트와 온라인 실기 평가를 통해 학생들의 프로그래밍에 어려움을 겪고 있는 내용들을 조사한다.

본 연구는 2020년 2학기 A 교육대학교 3학년 2개 학과(과학과, 사회과) 총 62명의 학생을 대상으로 적용한 ‘초등컴퓨터교육론’ 교과 중 프로그래밍에 관한 교육 사례이다. 초등 컴퓨터교육론의 구성은 초등정보교육과정과 교수법, 소프트웨어의 교과 활용, 프로그래밍 등이다. 이중 프로그래밍 교육에 관한 연구 내용으로 프로그래밍 교육방법, 프로젝트 학습, 마이크로 티칭, 실기 온라인 프로그래밍 평가, 성찰노트 작성, 그리고 학생들이 프로그래밍 과정에서의 어려움을 겪고 있는 사례에 대해 조사하였다. 수업은 코로나19로 인해 대면 수업이 불가능함에 따라 온라인으로 진행되었다.

## 2. 이론적 배경

### 2.1. 기존 연구

기존의 초등정보교육과 프로그래밍 교육에 관한 연구는 다양한 영역에서 이루어졌다.

정보교육과정에 관한 연구로서 안영희[1]는 초등 예비교원을 위한 교양수준의 정보교육과정(안)을 개발하였다. 그 구성으로 정보사회에서 발생하는 문제를 이해하기 위한 컴퓨터과학, 데이터관리 및 분석 등을 중심으로 문제 해결 능력 향상과 고등학교 정보과목에 대한 계속성과 계열성을 고려한 초등 예비교원을 위한 교육과정을 제안하였다. 양해지[13]는 초등교원 양성기관의 소프트웨어 융합 과목 현황분석을 통해 예비교사를 위한 융합교육 역량강화의 노력이 필요하다고 하였다. 유정수[14]는 현재 소프트웨어교육에 관한 시수가 매우 부족하며 이를

극복하기 위한 대학 평가와의 연계, 소프트웨어 교육에 대한 행·재정 지원 사업 확대, 학교 현장과 연계된 소프트웨어 교육 프로그램 확대, 초등교원 임용경쟁시험에 컴퓨터교육이나 소프트웨어 교육에 관련된 문제 출제 및 초등컴퓨터과목을 필수적으로 이수하도록 교원자격 검정을 개정하여야 한다고 주장하였다.

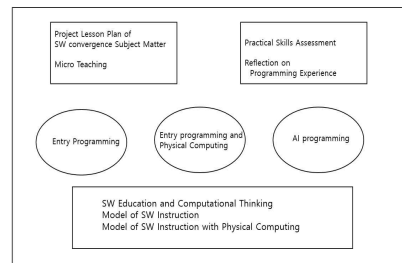
프로그래밍 교육의 연구로서 서현석[11]은 엔트리 프로그래밍을 활용한 협력적 이야기 생성하기 활동의 교육적 효과 분석을 통해 프로그래밍이 학생들의 소프트웨어 교육 역량에는 큰 변화가 없었지만 이야기 생성 능력은 통계적으로 유의미 하게 향상된 것을 조사하였다. 조미현[6]은 예비교사의 프로그래밍 학습경험과 소프트웨어 교육 과정 및 교육효과에 대한 이해와 실태 분석을 하였다. 그중 프로그래밍 학습 경험에서는 다수가 프로그래밍의 흥미, 유용성, 지속적 학습의 측면에서는 긍정적인 반면, 프로그래밍 경험에 만족하지 못하며, 프로그래밍이 어렵고, 자신의 프로그래밍 수준을 낮게 평가하는 것으로 밝혀졌다. 문외식[10]은 예비교사들이 프로그래밍 학습시 발생시키는 오류 데이터 분석을 통해 논리 오류가 가장 많았고, 문법 등의 사용미숙, 오타 등의 단순오류는 파이썬이 스크래치보다 매우 많은 것을 발견하였다. 이철현[9]은 초등 예비교사들이 겪는 EPL 프로그래밍의 학습의 어려움에 관한 연구에서 학생들은 문제분석 및 분해에 어려움을 겪고 있는 것을 발견하였다. 반면에 코딩의 어려움 인식은 상대적으로 낮게 나타났으며 또한 디버깅에 대해서도 어려워하는 것을 발견하였다. 초등예비교사에게 적용한 프로그래밍 교육의 사례로 한영신[5]은 비전공자가 프로그래밍 교육에서 겪을 수 있는 어려움을 고려한 새로운 프로그래밍 교육과정을 설계하여 실시한 결과, 프로그래밍 능력이 향상되는 것을 발견하였다. 또한 프로그래밍을 통한 컴퓨팅 사고가 쉽다는 의식 생성, 프로그래밍 언어에 대한 쉬운 접근, 프로그래밍 언어에 대한 관심도 향상됨을 밝혔다. 콜링[7]은 코딩 초보자들이 처음 코딩하는 방식은 코드의 분석, 설계, 테스트 등의 전체적인 공학적인 관점이 아니라 세부적이고 미시적으로 ‘프로그램의 한 줄, 한 줄’ 방식으로 하고 있음을 발견하였다. 또한 초보자들은 에러 발생 시 단순히 디버깅만 하는 수준으로 학습하므로 코딩 능력 향상에 한계가 있다고 하였다.

### 3. 연구방법

#### 3.1. 연구개요

본 연구의 구성은 A 교육대학교에 개설된 초등컴퓨터교육론 교과외의 프로그래밍 교육에 적용된 교육 방법, 학생들의 프로젝트 과제, 마이크로 티칭, 실기 평가, 성찰 노트, 학생들의 프로그래밍의 어려움 등에 관한 조사 등이다. 수업의 방식은 대학의 이러닝 시스템인 E-Class 시스템을 사용하였다. 교육 방법은 프로그래밍 교육 콘텐츠의 제공과 Zoom을 통한 실시간 수업 등 두 가지를 병행하였고 평가는 실기 평가로서 온라인으로 실시하였다.

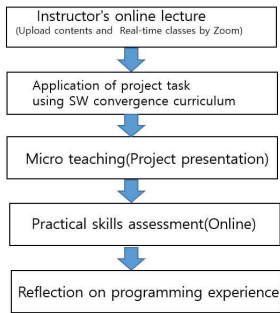
(Fig. 1)은 프로그래밍 교육과정의 구성도이다. 첫째, 코딩의 목적과 문제 해결방안으로 ‘소프트웨어교육과 컴퓨팅사고’이다. 둘째, 소프트웨어 활용이나 코딩을 교과에 융합 활용이다. 셋째, 문제해결을 위한 엔트리 프로그래밍, 피지컬 컴퓨팅, AI 프로그래밍의 실습이다. 넷째, 교과융합 소프트웨어 활용에 대한 프로젝트 과제수행과 마이크로 티칭 발표이다. 다섯째, 학습한 내용의 실시간 온라인 평가와 성찰노트이다.



(Fig.1) Configuration of Programming Curriculum

#### 3.2 교육과정

교육과정의 흐름은 (Fig. 2)와 같이 강사의 온라인 강의, 예비교사의 소프트웨어 교과융합교수 학습지도안 작성의 프로젝트 수행, 마이크로 티칭, 온라인 평가와 성찰 노트 등의 순서로 진행되었다.



(Fig. 2) Flow of Research

교육과정의 시수는 한 학기 1주당 2차시 분량으로 12주 총 24차시로 진행하였다. 사용된 교재는 엔트리의 '따라 하며 배우는 엔트리 프로그래밍\_초급'[2]를 주로 활용하였고, 데이터 과학의 부분은 엔트리의 교육자료 블로그의 자료를 응용하여 사용하였다[3]. 프로그래밍 교육과정 중 마지막 2주 동안에는 그때까지 학습한 내용을 기반으로 교과 융합의 소프트웨어 교수학습과정을 작성하는 프로젝트 과제를 수행하고 이를 마이크로 티칭 방법으로 발표하였다. 평가를 위해 서술평가 60점 만점에 35점은 초등컴퓨터교육론에 관한 단답형, 서술형 평가를 시행하고, 25점은 두 문제의 프로그래밍 실기 평가 문항을 출제하였다. 또한 예비교사 자신의 프로그래밍과 교과활용 수업 경험에서의 단점과 개선점을 찾기 위한 성찰 노트를 작성하였다.

### 3.3 프로그래밍 수업

수업의 방법은 시연중심 모형으로 강사는 학습 목표를 알려주고 그 내용을 시연중심으로 학생들이 모방, 제작하는 단계를 거치도록 하였다.

(Fig. 3)은 강의 화면의 예로서 '톤과 젤리'의 제목의 학습으로 주요 학습목표는 난수 활용하기, 변수 생성, 다른 오브젝트와 충돌하였을 때 점수를 증가하기 등이다.



(Fig. 3) Example of Lesson

### 3.3.1 시연

시연 단계에서는 학생들이 학습목표를 달성하기 위해서 다음의 단계를 수행한다. 첫째, 학습 목표를 달성하기 위한 분석단계로 구현에 필요한 블록 찾아내기 등을 수행하였고, 둘째, 문제 해결을 위한 블록 구성하기, 셋째, 에러 발생 시 문제를 찾아내서 고치는 디버깅하기 등의 연습을 반복하였다.

### 3.3.2 모방단계

시연 단계에서 학습한 내용을 응용하는 단계이다. 약간의 응용문제를 제시하여 학생들의 사고를 확장시켜준다. (Fig. 3)의 모방단계는 변수의 개념에 대한 학습의 내재화를 촉진시켜주기 위해 현재의 충돌 오브젝트 2개가 3개로 될 경우 총점수의 합을 어떻게 구할 것인지 등의 응용하는 문제로 진행하였다.

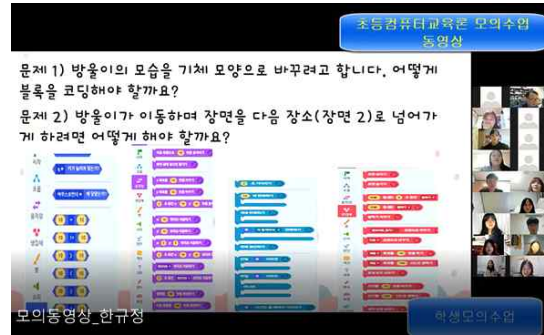
### 3.3.3 제작단계

제작 단계에서는 학습 내용을 기반으로 응용 수준이 높아진다. 따라서 제작단계에서는 시연, 모방 단계의 유사한 문제의 난이도를 높인 문제를 해결하도록 유도하였다, (Fig. 3)의 제작단계의 예로 충돌 점수가 5점 이상일 경우 2점을 증가시키는 프로그램 제작을 제시 하였다. 이 프로그램에서는 조건에 따라 변수의 값을 증가시키는 능력을 향상시키는데 그 목적이 있다.

### 3.4 프로젝트 수행

프로젝트의 과제는 수업과정에서 학습하였던 프로그래밍, 소프트웨어 활용 등을 교과에 융합 활용하는 것을 목적으로 한다. 과제의 산출물은 교수학습 지도 세안 1차시, 15분 분량의 마이크로 티칭 발표 자료 그리고 3분이내의 프로그램 실행 동영상이다. 과제의 수행은 2인 1 모둠으로 진행하였고 <Table 1>은 2개 전공학과(수학, 사회) 총 62명의 31개 모둠 과제의 과목별 수행 목록이다. 학생들에게 교과를 자유롭게 선택하도록 유도한 결과, 총 6개 교과 영역에서 31개의 프로젝트 과제가 진행되었다. 교과는 과학, 실과, 사회, 수학 과학 등의

선택이 많았는데 그 이유는 수강대상 전공학과 중 과학 전공학생들의 80%이상 과학교과에 소프트웨어 활용 지도안 작성을 선호하였는데 이는 자신의 교과에 소프트웨어융합 교육을 적용하려는 바람직한 시도로 분석된다. 다만 사회과 학생들은 과학과, 실과, 사회과 순으로 주제 선택이 많았다. 수행된 교과 프로젝트에는 모두 엔트리, 혹은 스크래치 등의 프로그래밍 언어를 사용하였고, 31개의 수행 프로젝트과제 중 9개 (29%)는 퍼지컬 컴퓨팅을 가미한 소프트웨어 융합 교육 지도안을 작성하였다.



(Fig. 4) Example of Micro teaching

<Table 1> Types of Projects  
count(%)

Subject Matter	Programming Utilization	Programming and Physical Computing
Science	16(52%)	4(44%)
Practical Arts	9(28%)	4(44%)
Social Studies	2(0.6%)	0(0%)
Mathematics	2(0.6%)	0(0%)
Music	1(0.3%)	1(11%)
Safety	1(0.3%)	1(11%)
Total	31(100%)	9(100%)

### 3.6 마이크로 티칭

학생들이 작성한 프로젝트과제의 수업으로의 실제 적용 가능성, 문제점 등을 확인하기 위해 모의 수업 환경의 마이크로 티칭을 진행하였다. 많은 과제수로 인해 모둠 당 15분 발표 및 5분 질의응답의 간이 마이크로 티칭으로 진행하였으며 진행방법은 강사의 과정 소개, 학생 발표, 학생들의 질의응답 및 토론, 강사의 지도 조언으로 구성하였다. (Fig. 4)는 온라인에서 학생들이 발표하는 화면으로 학생들은 다른 동료학생들이 초등학생이라고 가정하고 강의, 발문, 대답 등의 과정으로 소프트웨어 융합교육을 진행하였다.

토론은 대학의 LMS 시스템을 활용하였으며 토론시간을 효율적으로 활용하기 위하여 발표 도중이라도 경청하는 다른 학생들은 언제라도 토론게시판에 의문사항이나 코멘트를 동시에 올리도록 하였다. 또한 발표 모듈 중 발표하지 않는 다른 학생은 토론게시판을 빠르게 질의를 검색하여 발표 도중이라도 언제든지 응답하도록 하였다. 강사의 지도 조언은 학생들의 발표내용을 정리하고 미진한 질의응답에 대한 추가 설명으로 진행되었다.

### 3.7 평가

프로그래밍교육에서 가장 어려운 부분이 학생들의 프로그래밍 실기 능력의 평가이다. 특히 비대면으로 진행되는 평가는 여러 가지 어려운 점이 발생된다. 첫째, 학생들의 코딩능력을 측정하기 위한 객관식이나 단답형의 문항은 역량 평가에 한계를 가진다. 둘째, 온라인상의 코딩 평가를 허용하면 치팅 등의 부정행위를 탐지하기가 매우 어렵다.

따라서 본 연구에서는 다음과 같은 온라인 평가방식을 수행하였다. 평가 60점 만점에 35점은 초등컴퓨터교육론에 관한 단답형, 서술형 평가를 시행하고 이는 LMS 시스템의 평가 게시판을 통해 자동적으로 40분간 평가를 진행하였고, 25점은 두 문제의 온라인 프로그래밍 실기 평가 문항을 출제하였다. 실기문항은 (Fig. 5)와 (Fig. 6)과 같이 실기 문항과 동시에 문제의 실행결과에 대한 동영상상을 주어 문제의 이해를 도와주었다. 또한 평가 답안은 A4용지 2장 이내에 펜으로 작성하여 주어진 시간 내에 촬영하여 LMS 시스템에 과제 제출형식으로



음과 같은 방법을 적용하였다.

첫째, 문제 해결을 위한 학생들의 거시적 안목을 길러주기 위한 소프트웨어 교육 개론과 컴퓨팅 사고에 대한 기본 교육을 실시하였다.

둘째, 강의는 쉬운 프로그래밍 접근을 지향하는 시연 중심모형으로 구성하여 학생들은 따라 하기, 응용하기, 제작하기 등의 과정을 통해 코딩의 단계를 높이면서 응용문제에 대한 문제 역량을 쌓아나갔다.

셋째, 학생들은 소프트웨어와 교과 융합 프로젝트 과제로 2인 1 모둠으로 교수학습지도안을 작성하였고, 이런 프로젝트를 통해 소프트웨어를 교과에 융합하는 실재를 경험하였고 수업에 적용 가능성, 한계, 개선점을 알아보기 위해 발표, 토론 등을 포함하는 마이크로 티칭 실시하였다.

넷째, 수업과 프로젝트 과제를 통한 프로그래밍 실기 능력을 평가하기 위한 온라인 프로그래밍 평가로서 LMS 시스템의 평가기능과 과제 제출 기능을 활용하여 진행하였다. 이런 도구의 활용은 부정행위를 예방하는 공정한 평가가 진행되는데 효과적이었다.

다섯째, 프로그래밍 수업과 프로젝트 과제, 마이크로 티칭을 통해 자신의 소프트웨어 능력, 소프트웨어와 교과 융합 그리고 학습경험을 통한 긍정적인, 단점, 개선할 점 등의 수업 성찰 노트를 작성함으로써 프로그래밍 교육에 관한 학생 스스로의 다짐과 긍정적인 시야를 확립해주는 데 도움이 되었다.

31개의 수업 성찰지와 실기 평가를 통해 일부 학생들이 프로그래밍 과정에서 어려움을 겪는 몇 가지를 발견할 수 있었다.

첫째, 프로그래밍 과정에서 반복문에 대해 어려움을 겪는다. 특히 엔트리의 소스 코드 중에 많이 사용하고 있는 '계속 반복하기' 블록을 왜 사용하는지에 대한 의문을 많이 가지고 있었다. 학생들은 무한 반복에 대한 이해도가 낮아, 무한 반복을 사용할 경우와 아닌 경우에 대한 구별에 약점을 보였다.

둘째, 변수의 사용에 어려움 겪고 있었다. 문제가 주어지면 문제분석을 통해 필요한 변수를 설계하여 구현하여야 하는데 학생들은 이 부분에 대해 어려워하였다.

셋째, 학생들은 동영상 강의를 통해 프로그래밍 수업을 잘 이해를 할 수 있었지만 새로운 응용문제가 주어지면 어디서 어떻게 시작할지 막연하다는 의견을 제시

하였다. 시연 중심의 모형의 단점을 잘 지적하고 있었다,

## 5. 결론

본 연구는 A 교육대학교 초등 예비교사 3학년 학생 62명을 대상으로 2020년 2학기 '초등컴퓨터 교육론' 교과 내용 중 프로그래밍 교육의 수업사례이다.

교육과정은 대학의 이러닝 시스템에서 온라인으로 동영상 콘텐츠 제공과 Zoom 실시간 교육으로 진행하였다. 프로그래밍 교육방법으로는 초보자들에게 접근이 용이한 시연중심방법을 활용하였고, 학생들은 소프트웨어와 교과 융합의 교수학습지도안을 작성하는 프로젝트 과제를 수행하였다. 소프트웨어 교과 융합 과제의 적용 가능성과 한계, 개선점을 알아보기 위해 2주 동안의 마이크로 티칭을 통해 발표와 토론을 진행하였다. 또한 실기 평가로 코딩 실기 문항의 온라인 시험을 시행하였는데 부정행위를 방지하면서 공정하게 학생들의 코딩 능력을 평가할 수 있는 기회가 되었다. 또한 학생들은 성찰 노트 작성을 통해 자신의 프로그래밍과 소프트웨어 교과 융합 수업에 대한 장점, 단점 그리고 개선점을 되돌아보는 계기를 가질 수 있었다.

본 교육과정을 통해 학생들이 프로그래밍에 어려워하는 몇 가지를 발견할 수 있었다. 첫째, 코딩에서 '계속 반복하기' 블록 등의 무한 반복의 개념에 대한 이해가 낮았다. 둘째, 필요한 변수를 설계하여 구현하는 방법을 이해하는데 어려움을 겪고 있었다. 셋째, 학생들 동영상 강의를 이해하면서 코딩을 잘 따라 할 수 있었지만 막상 새로운 응용문제를 해결하려면 어디서 어떻게 시작할지 막연하다는 의견을 제시하였다.

앞으로의 연구로는 학생들이 어려움을 느끼는 부분에 대한 수준별 콘텐츠 제작과 실패를 극복하는 연습 방법 제시 그리고 강사와 학생들과의 활발한 상호작용 강화 방법 모색 등이 요구된다.



참고문헌

- [1] An, Y.H., Kim, J.M., Woo, H.S., Yang, H.J., Kim, M.J., Jung, D.Y., Lee, W.G.(2019). Development of Informatics Curriculum(Plan) for General Education Level in Prospective Elementary Teachers. *The Journal of Korean Association of Computer Education*, 22(1), 21-30.
- [2] Entry(2019). Follow-up and learn entry programming\_ Beginner. <https://play-entry.org/tt#!/basic/materials>
- [3] Entry(2020). Entry Blog. <https://blog.naver.com/entrylabs>
- [4] Han, K.J.(2016). Analysis of Computational Thinking Characteristics in Scratch Programs. *The Journal of Education Studies of Gongju National University of Education*, 53(2), 1-15.
- [5] Han, Y.S.(2018). Analysis of Effectiveness of Programming Learning for Non-science Major Preliminary Teachers' Development of Computational Thinking. *Journal of The Korean Association of Information Education*, 22(1), 41-52.
- [6] Jo, M.H.(2018). Analysis of Elementary Pre-service Teachers' Experiences and Understanding of Software Education. *Journal of The Korean Association of Information Education*, 22(1), 81-89.
- [7] Kolling, M., Rosenberg, J.(1996). Blue- A Language for Teaching Object-Oriented Programming. *Proc. of the 27th SIGCSE Technical Symposium on Computer Science Education*, 190-194.
- [8] Lee, C.H.(2019). Elementary School Teachers Difficulties in Learning Programming EPL. *Journal of Korean Practical Arts Education*, 32(2), 49-63.
- [9] Lee, J.H., Shim, J.K.(2020). An Analysis of Software Education Hours in Elementary School Teacher Training Institutions : Focusing on the Influence of SWEET Project. *Journal of The Korean Association of Information Education*, 24(2), 147-155.
- [10] Moon, W.S.(2018). Analysis of error data generated by prospective teachers in programming learning. *Journal of The Korean Association of Information Education*, 22(2), 205-212.
- [11] Seo, H.S., Jeong, Y.S.(2018). Educational Effects of Collaborative Story Creation Activities Using the Entry Programming Language. *Journal of The Korean Association of Information Education*, 22(6), 651-660.
- [12] So, M.H., Kim, J.M.(2016). Transference from learning block type programming to learning text type programming. *The Journal of Korean Association of Computer Education*, 19(6), 55-68.
- [13] Yang, H.J., Woo, H.S., Kim, S. H., Kim, H. N. Kim, J.M., Lee, G.W.(2020). Analysis of Software Convergence Curriculum in University of Education for Elementary Teachers. *Proceeding of The Korean Association Of Computer Education* 24(1), 51-54.
- [14] Yu J.S.(2018). A Study on the Development of Guidelines for Software Curriculum for Elementary School Educational Institutions. *Proceeding of The Korean Association Of Computer Education* 23(1), 311-342.

저자소개

한 규 정



1991 중앙대학교 대학원 컴퓨터공학과 (공학박사)  
 2010 미국 플로리다 주립대 교수설계학과 연구교수  
 2016 미국 샌버나디노대 수학 및 과학교육과 교환교수  
 1992~현재 공주교육대 컴퓨터교육과 교수  
 관심분야: 동기기반소프트웨어 설계, 인공지능과 퍼지컬 컴퓨팅  
 e-mail: kyujhan@gjue.ac.kr