

<http://dx.doi.org/10.17703/JCCT.2018.4.4.287>

JCCT 2018-11-36

EPL이 프로그래밍 교육의 프로그래밍, 컴퓨팅사고력 및 문제해결력에 미치는 영향에 관한 연구

A Study on the Effect of EPL on Programing, Computing Thinking and Problem Solving Ability of Programing Education

윤선희

Sunhee Yoon*

요약 본 논문은 비교적 기초학력 수준이 낮은 전문대학생들의 컴퓨터전공으로 프로그래밍 언어 교육을 수강하는데 있어서 알고리즘적 사고력과 컴퓨팅 사고력 및 문제해결력이 결여된 상태에서 프로그래밍 언어 교육을 받는데 현실적으로 어려움을 탈피하기 위해 EPL(Educational Programing Language)인 스크래치를 활용하여 코딩교육을 병행하여 프로그래밍 언어 학습을 수강한 학생들과 EPL을 활용한 코딩교육 없이 프로그래밍 언어 교육을 수강한 학생들의 프로그래밍과 컴퓨팅 사고력 및 문제해결력의 향상 정도를 측정하기 위해 학습효과분석, 문제해결 능력 향상 등에 대한 비교 분석을 실시하였다. 그 결과, EPL인 스크래치를 병행하여 프로그래밍 언어 교육을 수강한 학생들의 프로그래밍 및 컴퓨팅사고력과 문제해결력의 향상에 도움이 되는 것으로 나타났을 뿐만 아니라 학생들의 만족도 또한 높은 것으로 나타났다. 이것은 학생들이 어렵게만 생각한 프로그래밍실습에 자신감을 주었을 뿐만 아니라 중도탈락을 방지하는데 도움을 주는 것으로 나타났다.

주요어 : 교육용 프로그래밍언어, 스크래치, 프로그래밍언어 컴퓨팅사고력, 문제해결력

Abstract In this paper, it is practically difficult to obtain programming language education without having an algorithmic thinking ability, computing thinking ability, and problem solving ability of students with relatively low basic education. The results showed that students who took program language education in parallel with the EPL(Educational Programing Language), Scratch, compared to without using Scratch found to be helpful in improving their programming amd computing thinking ability, and problem-solving abilities as well as their satisfaction. This not only gave students confidence in their hard-thinking programming practices, but also helped prevent them from falling out of the middle.

Key words : Edicational Programing language(EPL), Scratch, Programing Language, Computing Thinking, Problem Solving

*정회원, 송의여자대학교 디지털미디어디자인과
접수일: 2018년 7월 22일, 수정완료일: 2018년 8월 11일
게재확정일: 2018년 9월 28일

Received: July 22, 2018 / Revised: August 11, 2018

Accepted: September 28, 2018

*Corresponding Author: shyoon@sewc.ac.kr

Dept. of Digital Media Design, SoongEui Women Univ, Korea

1. 서 론

‘서울 포럼 2017’의 주제로 4차 산업혁명 시대에 대비한 코딩 교육의 중요성이 강조되었다. 코딩은 컴퓨터 언어로 프로그램을 만드는 것으로 사물인터넷과 빅데이터 분석, 인공지능 등 4차산업과 관련된 모든 분야의 기초가 된다. 미국, 중국, 인도, 이스라엘, 영국과 일본 등 IT 강대국들은 이미 코딩을 정규 교육과정에 편입시켰고 우리나라도 2018년부터 전국 초·중·고교의 소프트웨어교육(코딩교육)을 의무화하고 있다. “21세기에는 코딩이 우리가 반드시 알아야 할 읽고 쓰는 능력이 될 것”이라며 “정규 교육과정에도 반드시 편성되어야 한다”는 의견이 제기 되었다[1]. <그림 1>은 코딩교육에 관한 초·중고등학생 및 학부모들의 의견을 조사한 결과이며 <그림 2>는 S/W 교육에 대한 관심을 나타내는 것으로 점점 S/W 교육의 중요성을 인식하고 대학교에서는 신입생을 대상으로 하는 S/W교육이 증가하는 추세임을 보여준다[1]. S/W 교육을 통해 학생들이 프로그램 언어 교육을 통해 추상적인 개념들을 이해하거나 자신이 해결하고자 하는 문제에 대한 해결 방법을 컴퓨터를 이용해 대안을 찾아낼 수 있으며, 컴퓨팅 사고력, 창의적 사고력 및 문제 해결 능력이 향상될 수 있다. 그러나 프로그램 교육을 처음 접하는 학생들에게 교육적 효과를 가지기 위해서는 프로그래밍 언어를 배우는 과정에서의 어려움을 극복해야 한다 이를 위해 최근에 교육용 프로그래밍 언어(EPL: Educational Programing Language)이 제안되었다.

2. 내년 중학교 입학생부터 소프트웨어(SW) 교육을 의무적으로 이수해야 하며, 초등학교는 2017년, 고등학교는 2018년부터 정식으로 SW 교과목을 편성합니다. 이에 어떻게 생각하십니까?

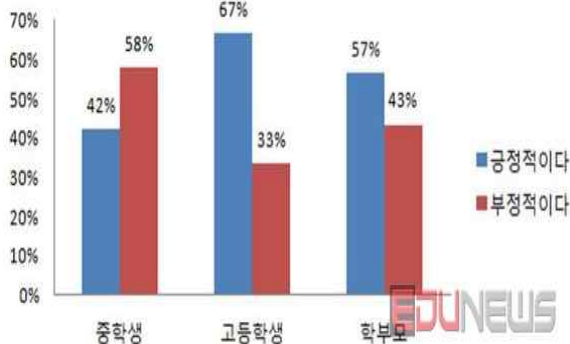


그림 1. 코딩교육에 관한 관심도[1]
Fig. 1. The Opinion of Coding Education



그림 2. S/W교육에 관한 추이
Fig. 2. Trend of S/W Education

EPL은 컴퓨터관련 전문지식이 필요 없는 초등학생이나 청소년들이 프로그래밍을 쉽게 할 수 있게 고안된 언어로서 교육적인 목적을 가지고 개발되었으며 비교적 용이하게 프로그램을 만들 수 있다는 장점이 있다. 또한 EPL은 알고리즘적 사고력과 문제해결력을 향상시키는 목적으로 하고 있으며 간단한 구문을 통해 학생들이 풍부한 상상력과 창의성을 표현할 수 있도록 개발되었다. <표 1>은 국내외에서 개발된 대표적인 EPL들이다.

알고리즘적 사고란 주어진 자료를 파악하고 각 단계마다 입출력되는 정보의 적합성을 판단하여 결과를 얻는 사고방식으로 복잡한 문제를 효율적으로 해결해 나가는 데 유용하다. 주어진 문제에 대한 알고리즘을 찾는다는 것은 주어진 문제를 해결하는 방법을 찾는 것을 의미하여 이 과정을 EPL을 활용하여 수행하는 것이 소프트웨어 교육의 핵심이라고 할 수 있다. EPL을 활용하여 수행하는 과정에서 컴퓨터가 이해할 수 있는 명확한 형태로 알고리즘을 표현할 수 있어야 하며 이 과정에서 컴퓨팅 사고력을 기를 수 있다. 컴퓨팅 사고력은 컴퓨터 해결 능력인 데이터 수집, 분석, 표현, 문제 분해, 추상화, 자동화 등을 사고에 적용시켜 다양한 분야에서 문제를 해결하는 데 필요한 능력이다[2]. 컴퓨팅 사고력은 단편적인 학습에서 탈피하여 복합적 사고로 확장해 가는 수단으로써 창의적 문제를 해결하는 핵심 능력으로 주목받고 있다.

본 논문에서는 비교적 기초학력 수준이 낮은 전문대 학생들의 컴퓨터전공으로 프로그래밍 언어 교육을 수강하는데 있어서 알고리즘적 사고력과 컴퓨팅 사고력 및 문제해결력이 결여된 상태에서 프로그래밍 언어 교육을 받는데 현실적으로 어려움을 탈피하기 위해 EPL을 활용하여 코딩교육을 수강한 뒤 프로그래밍 언어 학습을 수강한 학생들과 EPL을 활용한 코딩교육 없이 프로그램

언어 교육을 수강한 학생들의 컴퓨팅 사고력 및 문제해결력의 향상 정도를 측정하기 위해 학습효과분석, 문제해결 능력 향상 등에 대한 비교 분석을 실시하였다.

본 연구에서는 알고리즘을 이해하고 프로그래밍 및 컴퓨팅 사고력과 문제해결력을 이해하고 훈련하기에 적합한 EPL로 스크래치를 사용하였다. 스크래치는 문법이 간단하고 초보자도 쉽게 배울 수 있으며 기능에 적합한 컬러 블록들을 이동하여 문제를 해결하고 결과를 화면에 보여주는 비주얼 프로그래밍 도구이다. 스크래치를 통해 학생들의 동기 부여를 제공할 뿐만 아니라 소프트웨어 교육의 핵심인 창조적으로 사고하고 문제를 단계적으로 해결하며 다양한 방식의 문제를 접근할 수 있는 알고리즘적 사고력을 향상시킬 수 있다.

표 1. 교육용 프로그래밍언어 분석
 Table 1. Analysis of Education Program Language

분류	코딩언어	설명
국내	엔트리	KAIST에서 개발. 블록코딩 방식으로 소프트웨어 교육, 만들기, 공유와 토론, LMS가 가능한 통합형 소프트웨어 교육 플랫폼
	플레이봇	자바스크립트 학습 도구, 텍스트기반 코딩방식
	SiCi	한성대에서 개발. 블록코딩 방식으로 로봇교육에 특화된 교육 플랫폼
국외	스크래치	MIT에서 개발. 블록코딩 방식으로 손쉽게 게임, 애니메이션, 미디어아트 등 생성. 전세계적으로 가장 많이 사용.
	엠펙벤터	MIT와 구글이 공동개발 한 소프트웨어. 화면 디자인과 블록코딩 방식으로 안드로이드 앱 기반환경
	엘리스	미국의 카네기멜론대에서 개발하였으며 게임과 3D 애니메이션을 쉽고 재미있게 만들 수 있도록 고안.
	러플	파이썬을 이용한 교육용 프로그래밍 언어 환경으로 명령을 통해 화면에 있는 로봇을 움직이며 프로그래밍의 개념에 대해 학습.
	Zookeeper, Oozie	분산 환경에서의 자원제어/메타데이터 관리 기능

II. 관련연구

본 연구에서 활용한 EPL 소프트웨어인 스크래치는 MIT 미디어 연구소의 Lifelong Kindergarten Group과 UCLA가 컨소시엄에 의한 공동연구로 2007년에 개발된 교육용 프로그래밍 도구이다[3]. 스크래치는 학생들의 컴퓨팅 사고력 향상을 목적으로 컴퓨터 프로그래밍 기술의 향상뿐만 아니라 첨단 IT기술을 자율적으로 학습하고 이를 활용하여 창의적으로 문제를 해결하거나 프로젝트를 설계할 수 있는 능력향상을 위해 지속적으로 연구가 진행되고 있다.

1. 스크래치분석

스크래치는 무대, 스프라이트, 블록, 스크립트 영역으로 구분된다. 스크래치는 스프라이트라 부르는 다양한 객체들을 용이하게 정의할 수 있으며 스크래치의 프로그래밍 방식은 블록 영역에서 원하는 블록 요소들을 스크립트 영역에 드래그 앤드 드롭 방식으로 추가 또는 삭제한다. 블록 요소들은 동작, 형태, 소리 펜, 데이터, 이벤트, 제어, 관찰, 연산, 추가 블록 등으로 구성되며 색깔로 구분한다. 각 블록들은 레고와 같이 서로 끼워 맞출 수 있거나 안쪽에 또 다른 블록들을 넣을 수 있도록 구성되어 있으며 오류가 있으면 끼워 맞춰지지 않기 때문에 오류가 발생되지 않는 특징을 갖는다. 프로그램은 이벤트의 반응과 동작을 정의한 스크립트로 구성되어 있으며 무대 영역을 통해 스크립트를 확인할 수 있어서 프로그래밍이 직관적이다. 각 스프라이트는 다른 스프라이트와 독립적으로 작동한다. 스크립트는 이벤트 처리가 핵심이며 이에 따는 스프라이트의 동작이나 제어를 직관적으로 정의할 수 있다.

2. 스크래치 특징

스크래치의 특징은 첫째, 블록을 쌓는 놀이를 통해 학습하는 프로그래밍언어로 용이하게 코딩이 가능하며 10개의 블록영역에서 제공하는 블록들을 사용하여 쌓고 조립하는 것으로 프로그래밍이 가능하다. 둘째, 다양한 미디어의 삽입이 가능하여 사용자들의 흥미와 집중력을 향상시키며, 셋째, 확장성이 뛰어나서 데스크탑, 모바일, 태블릿 등, 다양한 장치에서 응용할 수 있으며, 공유성(shareability)이 뛰어나 프로그래밍 과정에서 다른 프로그램과의 제작내용이 공유가 가능하다, 다섯째,

한글을 포함한 다양한 언어를 지원하여 공유성을 증대시키고 사용의 편의성을 향상시킬 수 있다[4]. <그림 3>은 스크래치는 영역을 나타내는 것으로 8개의 특성을 지닌 컬러블록 팔레트, 스크립트, 모양, 소리를 표현할 수 있는 영역, 스크래치를 실행할 결과가 나타나는 무대영역 및 객체로 구성되는 스프라이트 영역으로 구성되어 있다. <그림 4>는 각 스프라이트에 대한 스크립트를 나타낸 예제이며 <그림 5>는 스크래치가 활용할 수 있는 피지컬 컴퓨팅 도구로 사용되는 보드이다.



그림 3. 스크래치 구성요소
Fig.3. Component Area of Scratch

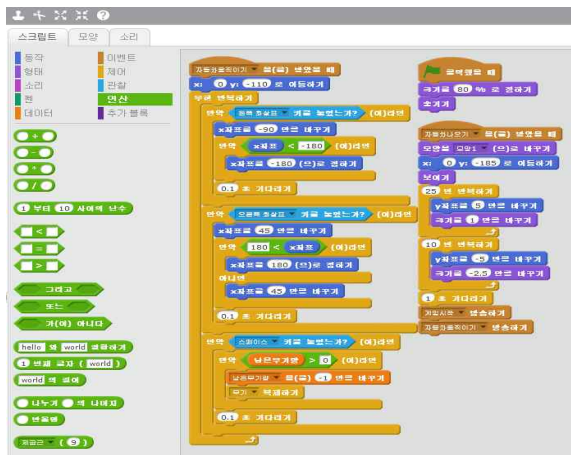


그림 4. 스크래치 스크립트예제
Fig.4. Example of Script of Scratch

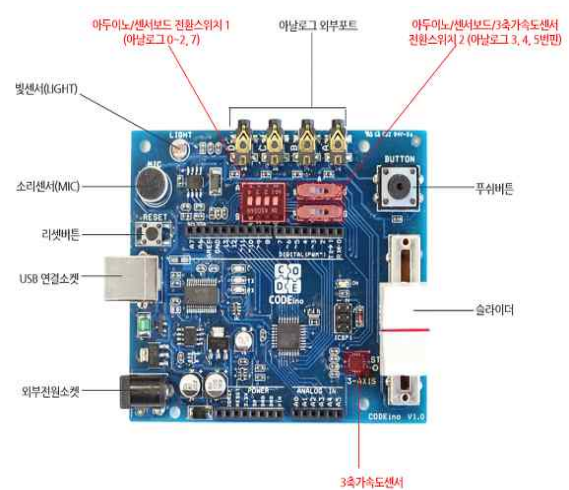


그림 5. 스크래치 활용 보드
Fig.5. Phsical Board for Scratch

III. 연구 내용 및 방법

1. 연구내용

대학 1학년 ‘프로그래밍언어실습’ 교과목을 수강하는 학생들 중 2개반을 대상으로 연구를 실시하였다. A반의 경우, 사전에 EPL수업을 받지 않은 학생들로 Java와 스크래치를 병행하여 수업 받는 학생들로 구성하였으며 B반의 경우, 사전에 EPL수업을 받은 학생이 3명이 있으며 Java 프로그래밍언어만을 수업 받는 학생들로 구성되었다. 이 수업들은 15주차 수업들로 이론과 실습이 병행되어지는 수업으로 진행되었다. 1주차 수업이 진행되기 전에 프로그램과 컴퓨팅사고력에 대한 사전 평가를 실시하였으며 15주차가 끝나면 후, 알고리즘과 컴퓨팅사고력에 대한 사후 평가 및 문제해결력에 대한 평가를 실시하였다. 문제해결력에 대한 평가는 학생들의 학습결과를 바탕으로 문제해결능력, 프로그램 작성 능력, 만족도를 평가하여 분석하였다. <표 2>는 프로그래밍 언어 관련 수업내용과 Java에서 진행되어야할 수업내용을 중심으로 이론수업과 실습수업을 구분하여 나타낸 예제이다.

대학 1학년 ‘프로그래밍언어실습’ 교과목을 수강하는 학생들 중 2개반을 대상으로 연구를 실시하였다. A반의 경우, 사전에 EPL수업을 받지 않은 학생들로 Java와 스크래치를 병행하여 수업 받는 학생들로 구성하였으며 B반의 경우, 사전에 EPL수업을 받은 학생이 3명이 있으며 Java 프로그래밍언어만을 수업 받는 학생들로

구성되었다. 이 수업들은 15주차 수업들로 이론과 실습이 병행되어지는 수업으로 진행되었다. 1주차 수업이 진행되기 전에 프로그램과 컴퓨팅사고력에 대한 사전 평가를 실시하였으며 15주차가 끝나면 후, 알고리즘과 컴퓨팅사고력에 대한 사후 평가 및 문제해결력에 대한 평가를 실시하였다. 문제해결력에 대한 평가는 학생들의 학습결과를 바탕으로 문제해결능력, 프로그램 작성 능력, 만족도를 평가하여 분석하였다. <표 2>는 프로그래밍 언어 관련 수업내용과 Java에서 진행되어야할 수업내용을 중심으로 이론수업과 실습수업을 구분하여 나타낸 예제이다.

표 2. 프로그래밍언어 관련 수업내용 예제
 Table 2. Class Contents of Programming Language

이론수업	
프로그래밍	- 학습 목표설정 - 수학적이고 논리적인 표현의 결과 평가 (정수형, 문자형, 실수형 등에 대한 우선순위) - 주어진 문제를 하향식으로 코드전환 (제어문의 개념, 구분 구조 해석, 제어문 활용에 대한 코드 전환 해석)
	실습수업
	- 다양한 데이터형 표현 - 프로그램 코딩 및 실행 - 연산자 적용 출력프로그램 작성 - 구조적 프로그램 이해 (순서, 선택, 제어문 구분 구조 이해)

<표 2>의 수업내용을 진행하는 과정에서 스크래치를 활용하는 A반의 경우, 스크래치의 제어팔레트의 구조를 이해하며 스프라이트를 활용하여 다양한 제어문의 구분 구조를 실습한 후, 스크래치를 활용한 실습에서 작성한 프로그램을 Java로 전환하는 과정이 진행되었다. <표 3>은 스크래치로 프로그램을 작성한 후, Java로 전환할 때 필요한 문법구조들을 비교분석한 것이다. <그림 6>은 3의 배수를 구하는 간단한 프로그램으로 스크래치를 활용하는 A반의 경우, 먼저 스크래치 프로그래밍을 학습한 후, Java 프로그램으로 전환하는 방식으로 수업이 진행된다.

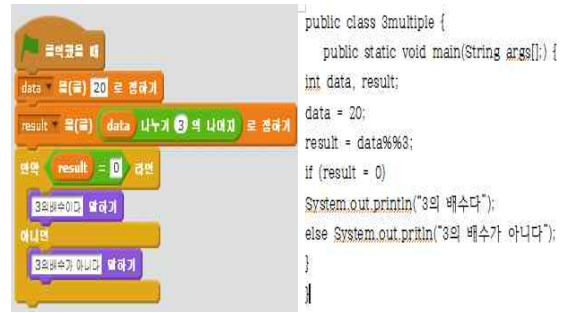


그림 6. 스크래치/자바 프로그래밍
 Fig.6. Scratch /Java Programming

표 3. Java/스크래치 문법 비교 예제
 Table 3. Example of Java vs. Scratch Programming

개념		Java	스크래치
순차	대입	i = i++; a = b+c;;	
	산술	+, -, *, /	
연산자	논리	&&, , !	
	관계	<, =, >	
	난수	random()	
	분기	if()	
제어문	if() - else	if() - else	
	반복문	for() while() do-while()	
변수	문자	char c;	
	정수	int i;	
	실수	float f; double d;	
배열(리스트)	배열 (Array)	int intarray[]; float farray[]; String sarray[];	

2. 연구방법

본 연구에서는 전문대 컴퓨터전공 여대생을 대상으로 스크래치를 활용한 Java 수업과 스크래치를 활용하지 않은 Java 수업을 들은 학생을 대상으로 15주차 수업을 진행하였다. 프로그래밍 및 컴퓨팅사고력 측정검사도구를 사용하여 사전평가와 사후 평가를 진행하였으며 스크래치를 활용반과 스크래치를 활용하지 않은 반의 Java 수업 15주차 종료한 후, 중간고사와 기말고사를 통해 학생들의 문제해결능력과 만족도의 결과를 분석하였다. 프로그래밍 및 컴퓨팅 사고력을 측정하기 위해 Hwang(2016)이 개발한 컴퓨터 활용능력검사 도구를 수정하여 사용하였다[2]. 컴퓨터 활용능력 검사도구는 프로그래밍 및 컴퓨팅사고력으로 분류되며 프로그래밍영역에는 프로그램 활용에 대한 자신감과 프로그래밍언어 학습에 대한 인지수준의 하위 영역으로 분류되어 각 하위영역에 3문항씩 6문항이며 컴퓨팅 사고력 영역에는 컴퓨팅 사고력에 대한 자신감 및 컴퓨팅을 통한 문제해결의 하위영역으로 구성되며 각 하위영역 별 3문항씩 6문항으로 총 12문항이 주어졌다<표 4>.

표 4. 프로그래밍/컴퓨팅사고력 측정도구 검사 항목
Table 4.Test Items of Programing/Computing Thinking

구분	영역	문항 내용
프로그래밍	프로그램활용에 대한 자신감	프로그램 사용법에 대한 숙지 정도
		프로그램으로 문제해결 할 자신감
		프로그램 활용 능력 정도
	프로그래밍 언어 학습에 대한 인지 수준	프로그래밍 학습의 난이도에 대한 사고
		EPL에 대한 관심 정도
		프로그래밍 학습의 필요성
컴퓨팅사고력	컴퓨팅사고력에 대한 자신감	논리적 설계 가능 여부
		문제를 프로그램을 통해 논리적 형태로 구조화
		일상생활의 문제를 알고리즘으로 표현
	컴퓨팅을 통한 문제해결	프로그래밍 언어를 활용하여 복잡한 구조 설계
		설계한 구조의 오류 수정
	복잡한 제어문의 프로그램 설계 능력	

문제해결력 검사도구는 Park(2012)가 사용한 평가 방법의 문항을 본 연구에 적합하게 수정 보완하였다[5]. 15주차 수업이 끝난 후, 중간고사와 기말고사를 포함한 평가와 실습을 통해 스크래치를 활용한 A반과 스크래치를 활용하지 않은 B반의 학습 결과 차이를 비교하여 분석하였다. 개념 및 문법의 이해를 측정하기 위한 영

역에는 7문항으로 구성되었으며 코드작성영역에는 4문항, 프로그램응용 영역에 2문항 등 총 13문항으로 구성되었다. 개념 및 문법의 이해 영역에는 0,1점의 값을 부여하였으며 코드 작성 영역은 5점을 부여했고 부분점수를 허용하였다, 프로그램응용 평가 영역은 실습시간의 과정 중에 작성한 프로그램으로 10점을 부여했으며 부분점수를 허용하였다<표 5>.

표 5. 문제해결력 평가 검사 항목
Table 5.Test Items of Problem Solving

구분	문제유형	비고
개념 및 구조 이해	변수선언문 이해	
	데이터형 이해	
	연산자 이해	
	제어문 이해	
	반복문 이해	
	배열 이해/선언문 작성	
	함수이해/선언문 작성	

코드작성	제어문 활용 프로그래밍	부분점수 허용
	반복문 활용 프로그래밍	
	배열 활용 프로그래밍	
	함수 활용 프로그래밍	
프로그램 응용	알고리즘구상	부분점수 허용
	프로그램 작성	

IV. 실험 및 결과

1. 프로그래밍/컴퓨팅사고력

본 연구에서 실험한 스크래치를 활용한 A반과 스크래치를 활용하지 않은 B반의 프로그래밍 및 컴퓨팅 사고력에 대한 사전 검사를 실시한 결과, <표 6>에서처럼 프로그램 및 컴퓨팅사고력에 대한 사전 검사는 스크래치를 활용한 A반의 경우, 평균점수는 8.96점이고 스크래치를 활용하지 않은 B반의 경우, 평균점수는 9.42점으로 A반과 B반간에 통계적으로 유의미한 차이가 나타나지 않았다.(P>.05).

표 6. 프로그래밍/컴퓨팅사고력 사전검사 결과
 Table 6. Pre-Test Result of Programing/Computing Thinking

구분	학생수	M	SD	t	df	p
스크래치활용반(A)	32	8.06	4.13	-1.654	59	.104
스크래치비활용반(B)	29	9.42	3.36			

15주차 후, 사후 검사결과는 <표 7>과 같이 스크래치를 활용한 A반의 프로그래밍 및 컴퓨팅사고력이 스크래치를 활용하지 않은 B반에 비해 유의미한 향상(p<.05)를 보이고 있으며<표 7>, <표 8>에 나타난 것과 같이 A, B반의 사전·사후 프로그래밍 및 컴퓨팅사고력 차이 검증 결과, 스크래치를 활용한 A반의 프로그래밍 및 컴퓨팅사고력의 향상이 통계적으로 유의미한 차이를 나타내고 있다.(p<.001). 이는 스크래치를 활용한 학습이 학생들의 프로그래밍 및 컴퓨팅사고력 향상에 긍정적인 영향을 미친 것으로 해석할 수 있다.

표 7. 프로그래밍/컴퓨팅사고력 사후검사 결과
 Table 7. Post-Test Result of Programing/Computing Thinking

구분	학생수	M	SD	t	df	p
스크래치활용반(A)	32	10.68	3.84	2.082	59	.014
스크래치비활용반(B)	29	9.88	3.38			

표 8. 프로그래밍/컴퓨팅사고력 사전·사후검사 결과
 Table 8. Previ vs Post Test Result of Programing/Computing Thinking

구분	학생수	사전문제해결 - 사후문제해결				
		M	SD	t	df	p
스크래치활용반(A)	32	-2.62	2.29	-6.82	31	.000
스크래치비활용반(B)	29	-.46	2.65	-1.53	28	.142

2. 문제해결력

<표 9>는 15주차 수업을 마친 후, 스크래치를 활용한 A반과 스크래치를 활용하지 않은 B반의 학습평가를 비교한 결과이다 이는 스크래치를 활용한 A반 학생들이 Java언어의 개념 및 문법에 대해 더 잘 이해하고 문제해결력을 통한 프로그램 작성과 이해에도 더 향상된 실력을 갖는다는 사실을 보여주는 결과이다.

표 9. 문제해결력 평가결과
 Table 9. Test Result of Problem Solving

구분	학생수	M	SD	t	df	p
스크래치활용반(A)	32	25.2	14	2.65	59	.0078
스크래치비활용반(B)	29	15.3	12.4			

15주차 수업을 종료한 후, 학생들이 스크래치 수업에 대한 만족도를 조사한 결과, 전체 문항에 대한 답변의 평균이 1-5점의 Likert 척도 중 4.18점으로 높게 나타났다. 이는 스크래치 프로그램이 학생들의 학습동기 유발에 효과가 있다는 것을 나타내며 Java 언어 프로그래밍 수업이 어렵다는 인식을 해소 시켜 중도탈락을 방지하는데 도움을 줄 수 있다는 것으로 해석할 수 있다. 또한 스크래치를 학습함으로써 Java문법을 이해하는데 도움을 주었을 뿐만 아니라 알고리즘적 사고력을 향상시키는데도 도움을 준 것으로 나타났다<표 10>.

표 10. 스크래치활용에 대한 만족도
 Table 10. Degree of Satisfaction using Scratch

구분	M	SD
코딩교육기간의 적정성	3.83	.828
코딩교육이 문제해결을 위한 알고리즘 사고에 미치는 영향	4.12	.745
코딩교육이 Java언어 수업의 관심에 미치는 영향	4.46	.618
코딩교육이 Java언어 문법의 학습에 미치는 영향	4.22	.684
코딩교육이 Java언어 프로그램 작성에 미치는 영향	4.27	.629
총계	4.18	.718

V. 결론

본 논문에서는 비교적 기초학력 수준이 낮은 전문대 학생들의 컴퓨터전공으로 프로그래밍 언어 교육을 수강하는데 있어서 알고리즘적 사고력과 컴퓨팅 사고력 및 문제해결력이 결여된 상태에서 프로그래밍 언어 교육을 받는데 현실적으로 어려움을 탈피하기 위해 EPL인 스크래치를 활용하여 코딩교육을 병행하여 프로그래밍 언어 학습을 수강한 학생들과 EPL을 활용한 코딩교육 없이

프로그래밍 언어 교육을 수강한 학생들의 프로그래밍과 컴퓨팅 사고력 및 문제해결력의 향상 정도를 측정하기 위해 학습효과분석, 문제해결 능력 향상 등에 대한 비교 분석을 실시하였다. 그 결과, EPL인 스크래치를 병행하여 프로그래밍 언어 교육을 수강한 학생들의 프로그래밍 및 컴퓨팅 사고력과 문제해결력의 향상에 도움이 되는 것으로 나타났을 뿐만 아니라 학생들의 만족도 또한 높은 것으로 나타났다. 이것은 학생들이 어렵게만 생각한 프로그래밍 실습에 자신감을 주었을 뿐만 아니라 중도탈락을 방지하는데 도움을 주는 것으로 나타났다. 고급의 Java 프로그램을 완전히 이해하는데 도움을 주기 위해서는 스크래치 활용의 폭을 확장해야 할 뿐만 아니라 다양한 EPL을 활용한 연구가 지속적으로 진행되어야 한다.

References

- [1] <https://blog.naver.com/hkw3339?Redirect=Log&logNo=220829601508>
- [2] Y, Hwang, K, Mun, Y, Park, “Study of Perception on Programming and Computational Thinking and Attitude toward Science Learning of High School Students through Software Inquiry Activity: Focus on using Scratch and physical computing materials”, Journal of The Korean Association for Science Education, Vol. 36, No. 3, pp. 325-335, 2016.
- [3] <http://scratch.mit.edu>
- [4] S, Cho., J, Song, S, Kim, K, Lee, “The Effect of CPS-based Scratch EPL on Problem Solving Ability and Programming Attitude”, Journal & Article Management System, KAIE, Vol. 12, No. 1, pp. 77-88, 2008.
- [5] J, Park, S, Cho, “The Effect of teaching Scratch in introductory programming course”, Journal of Policy Analysis and Management, Vol. 10, No. 9, pp. 449-456, 2012.
- [6] J, Song, S, Chos, T, Lee, Journal & Article Management System, KAIE, Vol. 12, No. 3, pp. 323-332, 2008.
- [7] H, Noh, S, Paik, “Student’s Perception of Scratch Program using High School Science Class”, Journal of The Korean Association for Science Education, Vol. 35, No. 1, pp. 53-64, 2015.
- [8] M, Lee, “Study on the Development of EPL Class Program Using Scratch”, Thesis of Master’s Degree, Seoul National University of Education, 2012.
- [9] J, Cho, Y, Lee, “The Development of Learning Program Using Scratch to Foster Logical Thinking Ability of Middle School Students in Technology Education”, Journal of Korean Technology Association, Vol. 12, No. 1, pp. 213-233, 2012.
- [10] T, Shin, “Development and Application of Teaching Learning Program for Elementary Science Using Scratch”, Thesis of Master’s Degree, Gyeongin National University of Education, 2018.
- [11] <http://blog.naver.com/sppaa3400?Result=Log&logNo=220963305960>
- [12] <https://m.post.naver.com/viewer/postView?volumeNo=14683015&memberNo=15460786&vType=VERTICAL>
- [13] J, Choi, E., Shin, S, Lim, “Content Restructure Model for Learning Contents using Dynamic Profiling”, JCCT, Vol. 4, No. 1, pp. 279-284, 2018