

CPS 모형 기반 스크래치 프로그래밍 학습이 언어 창의성에 미치는 영향

김병수[†] · 김종훈^{††}

요 약

최근 창의성의 영역 보편성 및 영역 특수성에 대한 논의가 계속되어지고 있는 시점에서 컴퓨터 프로그래밍 학습 활동 또한 창의성과 관련하여 다시 연구되어질 필요가 있다. 기존 교육용 프로그래밍 언어 학습의 창의성에 대한 측정을 TTCT 도형 검사를 사용한 것에 반해, 본 연구에서는 프로그래밍 활동이 언어적 사고 양식을 더욱 요구하는 창의적 활동이라는 관점으로 바라보고 있다. 이에 따라, 본 연구에서 컴퓨터 과학의 핵심 개념을 학습 내용으로 하는 창의적 문제 해결(CPS) 모형 기반의 스크래치 프로그래밍 학습을 개발하였다. 이 학습 프로그램을 4, 5학년 학생 17명을 대상으로 5일간 4차시씩 총 20차시 적용하고, 정규성이 검증된 이 실험집단에 TTCT 언어 검사의 평균 표준점수(100)로 t검정을 실시한 결과 언어 창의성의 세가지 하위 요인(유창성, 융통성, 독창성)과 창의성 지수(세 요인의 평균 표준점수)가 유의미하게 증진되었다.

주제어 : 창의적 문제 해결, CPS, 스크래치, 프로그래밍, 언어 창의성

Effects of Scratch Programming Learning based on CPS on Verbal Creativity

ByeongSu Kim[†], JongHoon Kim^{††}

ABSTRACT

Recently, the discussion of domain-general vs. domain-specificity of creativity has been continued. At this point in time, we need to research computer programming activities related to creativity again. While most of existing relative researches have performed TTCT figural tests for evaluating learners' creativity of learning education programming language, our perspective is that verbal creativity is needed on learning education programming language more. In this research, we have developed scratch programming learning based on CPS with the contents using fundamental concepts of computer science from the viewpoint of that programming is a kind of learning required verbal thinking style. This learning program was applied to 17 students of 4th and 5th grade for each 4 classes in 5 days, total 20 classes, this group passed normality test has the result of t-test has found that three subscales (fluency, flexibility and originality) and creativity index (mean of three standard scores) of verbal creativity were improved significantly using the mean of standard scores (100) of TTCT verbal tests as the test value.

Keywords : Creativity problem solving, CPS, Scratch, Programming, Verbal creativity

[†] 정 회 원: 제주대학교 대학원 컴퓨터교육전공 박사과정
^{††} 중신회원: 제주대학교 초등컴퓨터교육전공 교수(교신저자)
 논문접수: 2013년 08월 10일, 심사완료: 2013년 10월 11일, 게재확정: 2013년 10월 13일

1. 서 론

현 시대의 전 분야에서 창의성(창의력, 창의적 사고)은 빼놓을 수 없는 화두임이 분명하다. 창의성 관련 연구는 교육학, 심리학, 경영학, 공학, 디자인, 예체능 분야 등 다양한 분야에서 이루어지고 있지만, 가장 많이 이루어지고 있는 분야는 ‘교육계열’이다[1]. 컴퓨터 교육 분야에서도 창의성 신장을 위한 학습 프로그램 개발은 지속적으로 주목받는 연구 주제이다. 이 주제와 관련된 초반의 연구들은 프로그래밍 활동이 논리적 사고 및 문제해결력을 신장시키고 이에 부수적으로 창의성도 향상시킨다는 결과를 제시하고 있다. 하지만 이는 프로그래밍 활동의 어떠한 요인이 창의성 증진에 긍정적인 영향을 주었는지는 설명하지 못하고 있다[2].

이러한 이유로 최근 학습자의 창의성 증진을 주제로 CPS(Creative Problem Solving, 창의적 문제해결) 모형 기반의 프로그래밍 학습에 대한 연구가 활발해졌다[3][4]. 이 모형의 이론적 기반에서 창의성은 모든(또는 거의 모든) 영역에 보편적으로 적용될 수 있는 영역 보편성을 지니고 있다고 보고 있다. 즉, CPS 모형은 일반적인 대부분의 영역과 과제장면에 그대로 적용할 수 있다는 것이다[5]. 그럼에도 불구하고 “사고의 과정은 사고의 내용인 영역에 있는 지식과 얽혀있다.”, “발산적 사고나 동기가 요구하는 지식수준 등도 작업하는 내용과 과제의 수준에 따라 달라질 수밖에 없다”는 창의성의 영역 특수성 및 과제 특수성에 대한 반론은 간과되어서는 안 될 것이다[6][7]. 이는 프로그래밍 학습에서 CPS 모형을 적용하더라도 학습 내용 선정 및 내용 자체가 학습자의 창의성 신장에 매우 중요한 요소가 될 수 있음을 시사하고 있다.

이와 관련된 많은 연구들에서 창의성의 조작적 정의에 따라 창의성 증진을 측정하는 검사도구가 다양하게 사용되었다. 2001년에서부터 2012년까지의 국내 창의성 관련 연구에서 창의성 검사도구로 TTCT(도형 및 언어)가 가장 많이 사용되었다[1]. 컴퓨터 교육 연구 분야에서도 TTCT 검사가 주로 활용되고 있으나 대부분이 TTCT 도형검사를 하였다. 하지만 이 연구들에서 프로그래밍 활동이

TTCT 도형검사의 과제활동과 ‘사고의 양식’이 같기 때문임을 설명하고 있지는 못하고 있다 [8][9].

프로그래밍 학습에서처럼 학습자가 계산적 수행(Computational doing)이 필요한 문제해결의 사고 과정을 Wing은 계산적 사고(Computational thinking)라는 개념을 소개하며 설명하였다[10]. 계산적 사고에서 요구되는 추상화의 기술을 학습 시킨다는 것은 쉬운 일이 아닐 것이다[11]. 그럼에도 불구하고 컴퓨터 과학에서 프로그래밍 언어 학습에서의 핵심은 사실 추상화에 있다. 프로그래밍 활동에서 추상화의 학습은 단어와 기호의 조합으로 사용자의 심성모형을 코드로 바꾸는 문제해결 과정 자체에서 나타나게 되는데, 본 연구에서는 이러한 사고활동은 도형 창의성보다는 언어 창의성에 가깝다고 보고 있다.

본 연구에서는 CPS 모형 기반의 창의적 스크래치 프로그래밍 활동이 학습자의 언어 창의성에 어떠한 영향을 미치는지를 알아보기 위해 학습 프로그램을 개발하고 적용해 보았다.

2. 이론적 배경

2.1 언어 창의성

언어 창의성은 인간의 기존 지식과 경험을 기초로 하여 확산적 사고를 통해 다른 사람과 차별화된 독특하고 새로운 아이디어를 언어로 창출해내는 능력이라고 정의한다. 또한 언어 창의성은 체험, 상상, 생각한 바를 주제에 맞추어 논리적 문장으로 질서화 하는 행위으로써 고등 정신 능력을 세련시키는 사고기능의 최정점이라고 말하기도 한다. 그리고 이러한 언어창의성이 발휘될 수 있는 대표적인 영역이 글쓰기이다[12].

Moslemi는 언어 창의성의 하위요소로 독창성, 아이디어의 생성, 사용된 어휘, 글쓰기 스타일의 독특성을 제시하였으며[13], Malagdy와 Barcher는 언어 창의성의 구성요소를 생산성, 신기성, 융통성, 정교화, 비유적 언어의 사용으로 구분하여 제시하였다[14]. 언어 창의성의 측정변인에 대한 다양한 관점들이 대두되고 있지만, 본 연구에서는 창의성의 전통적인 창의성 구성요인들을 그 기준

으로 삼고 Torrance의 언어적 유창성, 융통성, 독창성을 하위척도로 하여 측정·분석하였고 언어 창의성의 사고 유형이 도형 창의성의 사고 유형과는 다름을 밝혔다[15].

2.2 프로그래밍과 언어 창의성

영어로 프로그램(program)의 어원적인 의미는 ‘문서를 짜기’(pro-gram), 즉 ‘미리 써보기’(writing-beforehand)란 의미이다. 프로그램의 개념과 실천은 바로 글쓰기이다. 우리가 흔히 쓰는 말로 프로그램을 ‘짜다’(to write)라고 하는 것이 글쓰기를 의미한다. 실제로 프로그램이란 종이 위에 쓴 것이 옮겨지든, 처음부터 컴퓨터에 쓰여지든 ‘글쓰기’라는 행위와 분리해서는 생각할 수 없다. 다시 말해 프로그램이란 말의 의미도 글쓰기이고 그 실천 행위도 글쓰기라는 의미이다[16]. 이런 의미에서 프로그래밍은 창의적인 글쓰기이며 논리적 문장으로 질서화 하는 행위인 언어 창의성을 요구하는 활동임에 틀림없다.

일반적으로 프로그래밍 자체는 문제를 해결할 수 있는 다양한 대안을 산출하고 최적의 해를 탐구하고 적용하는 과정에서 수렴적 사고를 요구하게 된다. 역설적으로 프로그래머는 이러한 과정에서 문제를 해결하기 위해 비판적·논리적 사고, 창의적 접근, 오류수정을 위한 다양한 시도를 해야 한다. 즉 프로그래밍은 수렴적 사고뿐 아니라 확산적 사고를 요구하는 특징을 가지고 있다[2].

Torrance는 ‘창의성 점수는 언어능력과 관계가 깊고 수학과는 관계가 높지 않다’고 했으며[17], Hargreaves는 ‘확산적 사고에 대한 연습이 언어 능력에 영향을 미친다’는 연구결과로 이를 지지하였다[18].

2.3 스크래치 프로그래밍

현재까지 전 세계적으로 개발된 교육용 프로그래밍 언어는 상당히 다양하다. 그 중 MIT공대, 미국과학재단, 인텔, UCLA 대학이 공동 제작한 스크래치의 장점은 사용자가 마치 레고 블록을 조립하거나 퍼즐을 맞추는 것처럼 그래픽 블록을 추가하는 것만으로 프로그래밍이 가능하여 어린

학생에서부터 성인까지 연령에 상관없이 사용이 쉽다는 장점이 있다[19]. 한글을 지원하며 최근 2.0 버전에서는 객체(스프라이트)의 복제, 함수 블록의 생성, 비디오 및 웹 클라우드 변수 지원과 함께 웹 브라우저 내에서 실행이 가능한 프로그램으로 업그레이드 되었다[20].

스크래치는 객체지향과 병렬처리 프로그래밍을 지원하는 프로그래밍에도 불구하고 사용자 입장에서 스크래치의 프로그래밍 인터페이스가 쉽다는 의미는 한편으로는 사용자에게 많은 추상화의 능력을 요구한다고 해석되어질 수 있다.

스크래치가 그래픽 기반의 사용자 환경을 제공하는 프로그램이기 때문에 텍스트 기반의 프로그래밍 언어에 비해 프로그래밍 언어 자체에 대한 암기를 줄이고 사용자의 흥미를 높인다는 장점이 있다. 이러한 장점은 사용자가 문제에 대해 더 탐구하고 해결책을 찾기 위한 사고 자체에 집중할 수 있다는 효율성을 동시에 가져온다.

2.4 CPS 모형

많은 연구들은 창의성을 ‘창의적 문제해결 과정’으로 본다. 창의적 문제해결 과정의 단계에 대한 연구 중에는 Hayes의 ‘완전한 문제해결자’와 Bransford와 Stein의 IDEAL이 있으며[21][22], 관련 연구들 가운데 가장 대표적인 것이 Osborn과 Parnes의 CPS이다[23][24]. Osborn과 Parnes의 CPS 모형의 5단계는 사실발견, 문제발견, 아이디어 발견, 해결발견, 수용발견으로 구성되어 있다.

사실 발견 단계에서는 문제를 파악하기 위한 정보와 자료를 수집하고 분석하는 단계로 객관적이고 정확한 역할이 필요하며 문제 상황에 대한 사실을 찾아야 한다.

문제 발견 단계에서는 다양한 관점으로 문제를 재정의하는 단계로, 해결하고자 하는 문제의 범위 안에서 핵심 문제 요소를 찾아 이를 간결하게 만드는 단계이다.

아이디어 발견 단계에서는 문제를 해결할 수 있는 다양한 대안들을 생각해 내는 단계로 가능한 한 많은 아이디어를 산출할 수 있도록 확산적 사고가 발휘될 수 있도록 돕는다.

해결 발견 단계에서는 아이디어 발견 단계에서

산출된 다양한 아이디어 중에서 최적의 아이디어를 고를 수 있도록 비교하고 평가하는 단계이다.

수용 발견 단계에서는 최적의 아이디어를 처음의 문제 상황에 적용하고, 실제 적용 시에 발생하는 방해 요인이나 제한점들에 대한 극복 방안도 함께 구안해야 하는 단계이다.

3. 학습 프로그램의 개발

3.1 내용의 선정

본 연구의 목적은 프로그래밍 언어 자체에 대한 습득이 아닌, 컴퓨터 과학의 핵심 내용들을 학습하며 언어창의성을 신장시킬 수 있도록 하는 것이다. 대부분의 스크래치 프로그래밍을 통한 창의성 연구에서의 학습 순서를 보자면, 몇 차시의 초반 수업에서 블록들의 기능을 먼저 다 익히고 중·후반 수업에서는 이를 기본적인 지식으로 창의적인 프로젝트를 설계하도록 함으로써 창의성을 신장하고자 하는 수업 전략을 가지고 있다. 이와는 달리 본 교재에서는 수업 전반에 걸쳐 블록들의 기능을 하나하나씩 배우게 되며, 새롭게 학습한 블록과 이전에 학습한 블록들의 조합을 통해 해결할 수 있는 문제가 제시되고 이 문제들을 해결해 나갈 수 있는 다양하고 창의적인 접근을 유도하게 된다.

본 연구에서는 CS(Computer science, 컴퓨터 과학) 핵심 개념에 대한 기존 연구를 참고하여 추출하여 제시하고[25], 이들의 학습 위계에 따라 <표 1>과 같이 순차적으로 학습 내용으로 제시하였다.

<표 1> 학습 순서

순서	CS 핵심 개념	학습할 핵심 블록
1	순차적 구조	동작 블록
2	반복	펜블록, 제어블록
3	조건, 분기	관찰블록, 형태블록
4	변수, 난수	데이터블록, 연산블록
5	동시성과 병렬처리	이벤트 '방송하기' 블록
6	객체, 시뮬레이션	스프라이트 변수
7	함수, 재귀	이벤트 '방송하기' 블록

3.2 창의 프로그래밍 문제의 제작

본 학습 프로그램 개발의 핵심은 창의성 요구 문제의 제작에 있다. 예를 들어 스크래치에서 블록 하나의 기능을 학습하는데 이것을 이용하여 다양한 접근으로 해결할 수 있는 문제를 제작할 수 있는지가 관건이다. 모든 프로그래밍 문제는 실제 문제를 해결할 프로그램을 만들기에 앞서 손으로 간단한 코딩을 할 수 있는 축소된 범위의 문제가 먼저 제시되어 이해도를 높이도록 하였다. 또한 프로그래밍 아이디어를 손으로 쓰거나 그림과 기호를 이용하여 프로그램을 설계할 수 있도록 하였다. 이러한 활동은 학습자들이 추상화에 더욱 익숙해지게 하며 스크래치의 블록 명령 또한 하나의 추상화된 약속임을 인지하게 한다[26].

창의성 요구 문제의 제시를 통한 CPS 모형 기반의 학습 과정은 <그림 1>과 같다.



<그림 1> 학습 과정

3.3 세부 학습 내용

CS 핵심 개념의 선정과 CPS 모형 기반의 학습 과정에 따라 실제 학습의 세부 내용은 <표 2>와 같다. 1차시의 수업을 40분 수업으로 하여 총 20차시의 수업으로 진행된다. 학습 초반에서는 프로그래밍을 처음 접하는 학습자들을 위해 손코딩의 기회를 많이 접하게 하여 프로그래밍의 개념과 감각을 익히게 하도록 하였다. 손코딩이란 문제를 해결하기 위한 아이디어를 기호, 그림 또는 단어나 문장으로 간략하게 추상화시키는 전략을 말한다[26].

<표 2> 세부 학습 내용

순서	차시	세부 학습 내용
1장. 동작과 순서		○ 스크래치 화면 구성 알기 ▶ 무대와 좌표, 동작 블록 익히기
	(2)	○ 창의적으로 생각하기 (손코딩 #1#2#3) ● 창의 프로그래밍 문제 - 버스 운전하기 #1#2
	(1)	▶ 펜 블록 익히기 ○ 창의적으로 생각하기 (손코딩) ● 창의 프로그래밍 문제 - 한 붓 그리기 #1#2 / - 우유 배달하기 - 외계인의 미스터리 서클 만들기
2장. 제어와 관찰, 형태		▶ 제어의 '반복' 블록 익히기 - 날 잡아봐라 / - 이어 달리기
	(3)	● 창의 프로그래밍 문제 - 계단 그리기 (3개의 블록만 사용하기) - 정사각형 (최소의 개수의 블록 사용하기) - 정다각형 그리기 - 미디어 아트 프로젝트 만들기
	(2)	▶ 제어의 '조건' 블록, 관찰/형태 블록 익히기 ● 창의 프로그래밍 문제 - 고양이를 쫓는 박쥐떼 만들기 - 스무고개로 숫자 놀이
3장. 변수, 연산과 동시 처리		○ 변수의 개념 이해하기 ▶ 데이터 블록 익히기
	(2)	● 창의 프로그래밍 문제 - 계산기 디자인하기 / - 변수 교환하기 - 3의 배수이며 5의 배수인 수 - 변수를 이용한 도형 만들기
	(2)	○ 난수의 개념 이해하기 ▶ 연산의 '난수' 블록 익히기 ● 창의 프로그래밍 문제 - 난수 그래픽 효과 / - 열고개 숫자 놀이 - 가위 바위 보 게임 만들기 - 아날로그 시계 만들기
4장. 객체 지향, 인공 지능 그리고 시물 레이션		○ 동시처리의 개념 이해하기 ○ 창의적으로 생각하기(동시처리의 문제들)
	(2)	● 창의 프로그래밍 문제 - 과녁 맞추기 게임 만들기
	(2)	○ 객체와 인공지능의 개념 이해하기 ▶ 데이터의 스프라이트 '변수' 블록 익히기 ● 창의 프로그래밍 문제 - 1부터 10까지 말하는 고양이 - 땅을 파는 인공지능 지렁이 - 장애물을 피하는 더듬이 벌레 - 청소로봇 #1#2
		○ 함수와 재귀의 의미 이해하기 ▶ 이벤트의 '방송' 블록 익히기 ● 창의 프로그래밍 문제 - 박테리아와 백신 / - 감기의 전염

3.4 실제 학습의 예

앞서 제시한 5단계의 학습과정에 따른 세부 학습 내용에 대한 전개를 살펴보자.

① 단계: 학생은 손코딩을 통해 컴퓨터과학에서의 프로그래밍의 처리 방법 이해한다. (그림 2, 3(좌))

길 안내하기

아래의 그림은 민수의 동네 약도이다. (하늘에서 아래를 내려다 본 그림)

▲: 박물관
●: 원위치

길을 가던 관광객이 민수에게 박물관까지 가는 길을 물었다. 민수는 관광객에게 아래와 같이 6줄짜리 안내문을 써주었다.

순서	동작
①	원위지에서 길찾기 시작
②	동쪽 방향으로 보기
③	4블록(간) 움직이기
④	북쪽 방향으로 보기
⑤	2블록(간) 움직이기
⑥	원쪽을 보면 목적지가 보임

(※ 안내문의 길: 6)

관광객은 목적지에 도착할 때까지 총 6번의 동작을 했고, 박물관을 잘 찾을 수 있었다.

생각해 봅시다.

아래의 그림은 우리 동네 약도이고 어떤 관광객이 나에게 미술관까지 가는 길을 물었다.

▲: 미술관
●: 원위치

미술관까지 가는 길은 다양하지만 나는 관광객이 쉽게 길을 찾아가길 수 있도록 안내문의 길이를 줄이고 싶었다. 민수의 방법처럼 안내문을 작성하되, 원위지에서 미술관까지 가는 길이가 가장 짧은 안내문을 완성하여야.

순서	동작
①	원위지에서 길찾기 시작

<그림 2> '창의적으로 생각하기'의 예(1장)

순서	동작
①	원위지에서 길찾기 시작
②	서쪽으로 방향보기
③	2블럭 가기
④	북쪽으로 방향보기
⑤	1블럭 가기
⑥	동쪽으로 방향보기
⑦	4블럭 가기
⑧	오직을 보면 목적지 보임

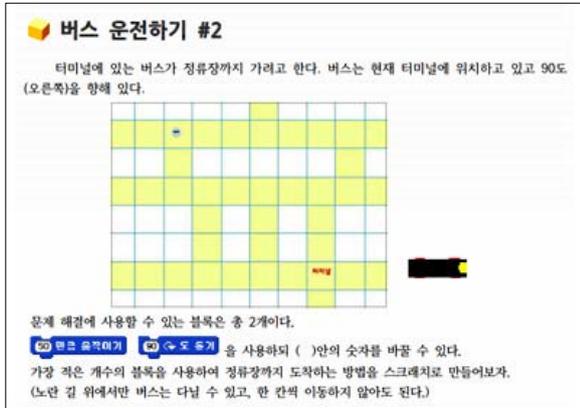
책상 스티커이벤트와 책상 문구대장 만들기

책상은 컴퓨터이벤트나 컴퓨터이벤트
- 컴퓨터이벤트 - 컴퓨터이벤트
- 컴퓨터이벤트 - 컴퓨터이벤트
- 컴퓨터이벤트 - 컴퓨터이벤트

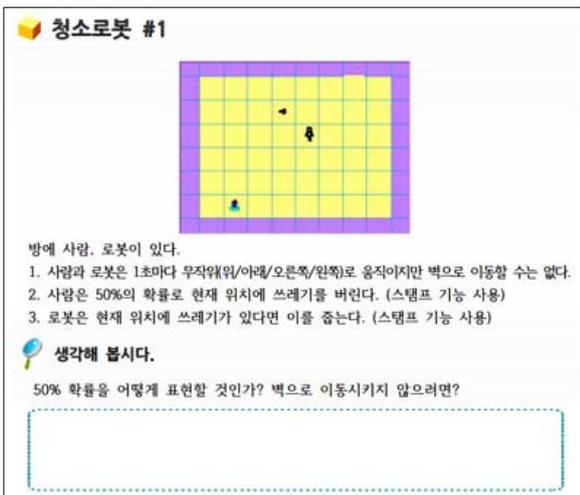
① 컴퓨터이벤트 - 컴퓨터이벤트
② 컴퓨터이벤트 - 컴퓨터이벤트
③ 컴퓨터이벤트 - 컴퓨터이벤트
④ 컴퓨터이벤트 - 컴퓨터이벤트
⑤ 컴퓨터이벤트 - 컴퓨터이벤트
⑥ 컴퓨터이벤트 - 컴퓨터이벤트
⑦ 컴퓨터이벤트 - 컴퓨터이벤트
⑧ 컴퓨터이벤트 - 컴퓨터이벤트

<그림 3> 학습자의 손코딩(좌) 및 아이디어 메모(우)

②단계: 교사는 손코딩의 문제와 유사한 스크래치 프로그래밍 문제를 프로젝트로 제시한다(그림 4, 5).



<그림 4> '창의 프로그래밍 문제'의 예(1장)



<그림 5> '창의 프로그래밍 문제'의 예(4장)

- ③단계: 학생들은 주어진 문제의 해결을 위해 다양하게 탐색한다. 물론 스크래치블록을 이용하여 해결할 수 있도록 고안해야 하며, 이 때 아이디어를 손코딩으로 메모한다(그림 3(우)).
- ④단계: 실제 스크래치로 다양한 아이디어를 테스트해보며 최적의 해답을 찾는다.
- ⑤단계: 문제의 해답을 프로젝트로 완성하여 다른 이가 실행할 수 있도록 한다.

'창의적으로 생각하기' 문제를 통해 학습자는 생활에서 밀접한 문제를 접하게 되고, 직접 손코딩을 통해 문제를 해결할 수 있는 기회를 갖게 된다. 이를 통해 프로그래밍의 기본적인 규칙을 익히게 되며 다양한 해결 방법을 찾으려는 확산적 사고를 유도할 수 있다. 학습자의 해결 방법의 정당유무는 짝이나 교사가 직접 확인한다.

이러한 활동을 통해 학습자는 '창의 프로그래밍 문제'에서 컴퓨터 과학의 기본 개념을 갖고 문제에 접근하게 된다. 문제 자체에 학생들이 집중할 수 있도록 스프라이트를 직접 그리며 만드는 활동들은 최소화하고 미리 제작한 문제 상황의 프로젝트를 제공하도록 한다.

4. 연구의 방법

4.1 연구 가설

본 연구의 가설은 다음과 같다.

- 영가설: CPS 모형 기반 스크래치 프로그래밍 학습에 의한 학습자의 언어창의성은 차이가 없다.
- 대립가설: CPS 모형 기반 스크래치 프로그래밍 학습에 의한 학습자의 언어창의성은 차이가 있다.

4.2 연구 대상 및 실험 설계

본 연구는 교육기부 프로그램으로 소개되어 제주도 초등 4, 5학년 학생들 중 선착순 지원자 17명(4학년 12명, 5학년 5명)을 대상으로 총 8일 동안 진행되었다. 전체 교육기부 프로그램의 오리엔테이션과 사전·사후의 TTCT 언어 검사와 최종 결과 발표회를 제외하면 본 연구에서 개발된 프로그램은 5일간 4차시씩 총 20차시 수업으로 진행되었다. 강사 1인이 전체 학습을 진행하였고 보조 강사 2인은 손코딩 및 프로그래밍 활동에 도움을 주었다. 모든 학생들은 데스크 탑 컴퓨터를 사용하였고 윈도우용 스크래치 버전 1.4를 사용하였다.

4.3 검사 도구 및 방법

본 연구의 검사도구는 Torrance가 개발한 TTCT 언어 검사(김영채의 한국 번역판)로 사전 검사에 A형, 사후검사에 동형 검사지인 B형을 투입하였다[15]. TTCT의 언어검사는 6개의 과제(활동)로 구성되어 있고 이를 통해 창의성의 3가지 하위 요인(유창성, 융통성, 독창성)을 측정할 수 있다. 채점은 창의력 한국 FPSP가 인정하는 TTCT 채점 전문가를 통해서 이루어졌다.

TTCT는 많은 사례의 표본 검사를 통해 요인별, 학년별 통계치가 이미 제시된 타당도·신뢰도

가 높은 검사이다. 본 검사의 기준표의 표준점수들은 평균치가 100, 표준편차가 20인 분포를 이루고 있기 때문에 본 연구에서 얻은 검사 결과는 비교집단 없이 단일 표본 검증을 실시하면 될 것이다. 하지만 실험집단이 소규모인 관계로 결과값에 대한 모수/비모수 통계를 결정하기 위해 일표본 콜모고로프-스미르토프 검정을 사용한 결과 ‘실험집단의 창의력 해당 요소에 대한 값 평균이 100, 표준편차가 20인 집단이다’라는 귀무가설에 대해 유창성(유의확률 .619), 융통성(유의확률 .603), 독창성(유의확률 .318), 창의력 지수(유의확률 .409)의 유의확률이 $p < .05$ 에서 모두 기각하지 못하여 실험집단의 정규분포를 확인하였다. 이에 따라 창의성 각 요소에 대해 일표본 t검정을 사용하였다.

5. 연구의 결과

5.1 연구 결과와 해석

먼저 사전 검사에 대해 일표본 t검정을 실시하였다.

<표 3> 사전 t검정 결과

요인	평균	편차	t	p
유창성	98.71	13.665	-.390	.701
융통성	101.88	14.628	.531	.603
독창성	103.00	15.540	.796	.438
창의력 지수	101.24	14.087	.362	.722

실험집단의 유창성(98.71)의 평균은 모집단의 평균 이하였고, 융통성(101.88), 독창성(103.00), 창의력 지수(101.24, 평균 표준점수)의 점수는 모집단의 평균(100)보다 높았지만 유의미한 차이는 보이지는 않았다. 이는 본 연구에 참여한 학습자의 집단이 모집단인 4, 5학년 학생들의 집단과 같음을 의미한다고 볼 수 있다. 본 연구에서 개발한 학습을 적용한 후에도 일표본 t검정을 실시하였다.

<표 4> 사후 t검정 결과

요인	평균	편차	t	p
유창성	111.71	13.499	3.575	.003
융통성	114.06	11.239	5.158	.000
독창성	111.71	13.499	3.575	.003
창의력 지수	111.82	12.300	3.964	.001

사후 검사 결과로 유창성은 111.71, 융통성은 114.06, 독창성은 111.71, 창의력 지수는 111.82로 t검정에 대해 유의미한 차이를 보였다($p < .05$). 이는 사후의 실험집단이 모집단의 평균(100)보다 유

의미하게 창의력 요인과 창의력 지수가 증진되었음을 의미한다고 볼 수 있다.

<표 5>의 실험집단 내 사전·사후의 창의력 요인 및 창의력 지수에 대한 대응 표본 t검정의 결과 또한 앞서의 일표본 t검정에 대한 결과를 뒷받침하고 있다.

<표 5> 사전·사후의 대응 표본 t검정 결과

사전-사후	상관 (p)	평균	t	p
유창성	.693(.002)	-11.118	-4.312	.001
융통성	.543(.024)	-12.176	-3.946	.001
독창성	.643(.005)	-8.706	-2.892	.011
창의력 지수	.658(.004)	-10.588	-3.958	.001

사전·사후의 각 요인과 창의력 지수는 서로 상관관계가 높게 나타나 서로 종속되어 있음을 보여주고 있으며 대응 표본 t검정 결과 모든 요인과 창의력 지수에 유의미한 차이를 보였다($p < .05$).

앞선 일표본 t검정과 대응 표본 t검정의 결과로 연구 가설인 영가설은 기각되어, 본 연구에서 개발한 CPS 모형 기반 스크래치 프로그래밍 학습에 의한 학습자의 언어창의성은 증진된다고 결과를 해석할 수 있을 것이다.

6. 결론

‘창의’, ‘창조’라는 단어가 교육·사회·문화의 패러다임을 바꿔 놓고 있을 정도로 대세이다. 컴퓨터 교육 연구 분야에서도 프로그래밍 학습에 의한 창의성 신장을 결과로 이끌어낸 연구가 많았지만 전략이 세부적이지 못했다.

본 연구에서는 프로그래밍이 언어적 사고양식을 필요로 하는 활동이며 언어 창의성과 관련 있다고 판단하고, CPS 모형 기반의 학습 프로그램을 개발하고 적용하였다. 또한 학습 내용 자체가 창의성을 요구하는 문제가 되기 위해 컴퓨터 과학의 핵심 개념을 주제로 손코딩 및 창의 프로그래밍 문제를 제작하여 적용하였다. 본 연구의 결과로 학습자의 언어 창의성의 모든 하위 요인과의 평균 표준점수인 창의력 지수를 유의미하게 증진시켰음을 보여주었다.

본 연구 결과를 통해 프로그래밍 활동이 언어 창의성을 증진시킬 수 있는 언어적 사고 양식을 요구하는 활동이라는 점은 이후 관련 연구에 영향을 줄 수 있을 것이라고 본다.

참 고 문 헌

- [1] 정미인, 정혜인, 정세영, 김영채 (2013). 2001-2012년 창의력관련 연구의 통합적 분석. **사고개발**, 9(1), 1-26.
- [2] 전성균, 서영민, 이영준 (2011). 창의성과 프로그래밍 교육에 관한 고찰. **컴퓨터교육학회 학술발표대회논문집**, 15(1), 73-77.
- [3] 전성균, 이영준 (2012). 초등학생의 확산적 사고 촉진을 위한 CPS 프로그래밍 수업의 효과 분석. **컴퓨터교육학회논문지**, 15(2), 1-8.
- [4] 윤선희, 허민, 진영학, 김영식 (2010). 중학교 “정보” 교과 “문제 해결 방법과 절차” 영역의 CPS 모형을 적용한 수업 설계. **컴퓨터교육학회 학술발표대회논문집**, 14(1), 13-18.
- [5] 김영채 (2012). 창의력의 영역 보편성과 특수성: 쟁점과 TTCT 창의력 검사의 분석. **사고개발**, 8(1), 1-29.
- [6] Willingham, D. T. (2008). Critical thinking: Why is it so hard to teach?. *Arts Education Policy Review*, 109(4), 21-32.
- [7] Baer, J. (2011). Why teachers should assume creativity is very domain specific. *International Journal of Creativity & Problem Solving*, 21(2), 57-61.
- [8] 이태욱 (2006). **마이크로 로봇 교육을 통한 초등학교 창의성 계발에 대한 연구**. 석사학위 논문, 제주교육대학교.
- [9] 이민희 (2009). **두리틀을 이용한 프로그래밍 수업이 창의성, 문제해결력, 프로그래밍 흥미도 향상에 미치는 영향**. 석사학위 논문, 전주교육대학교.
- [10] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- [11] Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, 50(4), 36-42.
- [12] 강정란 (2008). **명사-명사 결합개념을 이용한 글쓰기활동이 아동의 언어 창의성 및 쓰기 표현력에 미치는 효과**. 석사학위 논문, 부산대학교.
- [13] Moslemi, M. H. (1977). The grading of creative writing essays. *Research in the Teaching of English*, 9, 154-161.
- [14] Malgady, R. G., & Barcher, P. R. (1979). Some information-processing models of creative writing. *Journal of Educational Psychology*, 71, 717-725.
- [15] 김영채 (2010). **검사요강: Torrance TTCT (언어) 검사 A&B형**. 대구: 창의력 한국 FPSP.
- [16] 장승권 (1996). 정보기술의 발전과 글쓰기로서의 컴퓨터 프로그래밍. **문화과학**, 9, 51-68.
- [17] Torrance, E. P. (1963). *Education the creative potential*. Minneapolis: University of Minnesota Press.
- [18] Hargreaves, D. J. (1977). Sex roles in divergent thinking. *British Journal of Education Psychology*, 47, 25-32.
- [19] 김수환, 이원규, 김현철 (2009). 개정된 정보교육과정에서 교육용프로그래밍언어의 교육적 적용방안. **컴퓨터교육학회 논문지**, 12(2), 23-31.
- [20] Lifelong kindergarten in MIT Media Lab. (2013). What's New in Scratch 2.0?. Retrieved from <http://scratch.mit.edu/overview/>
- [21] Hayes, G. R. (1981). *The complete problem solver*. Philadelphia: Franklin Institute Press.
- [22] Bransford, G. D., Stein, B. S. (1984). *The ideal problem solver*. New York: Freeman.
- [23] Osborn, A. F. (1957). *Applied imagination*. New York: Scribner.
- [24] Parnes, S. J. (1967). *Creative behavior guidebook*. New York: Charles Scribner's Sons.
- [25] Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. M. (2010, August). Learning computer science concepts with scratch. In *Proceedings of the Sixth international workshop on Computing education research* (pp. 69-76). ACM.
- [26] 김병수, 김종훈 (2012). 초등 예비교사의 컴퓨터과학에 대한 인식 변화를 위한 계산적 사고 기반 알고리즘 학습의 설계 및 적용. **한국수산해양교육연구**, 24(4), 528-542.



김 병 수

2010 제주교육대학교
컴퓨터교육과(교육학학사)
2010 제주대학교
컴퓨터교육과(교육학석사)

2003~현재 초등학교 교사
2010~현재 제주대학교 컴퓨터교육전공 박사과정
관심분야: Computational Thinking, EPL 교육
E-Mail: pigpotato79@naver.com



김 종 훈

1998 홍익대학교
전자계산학과(이학박사)
1998~1999 한국전자통신연구원
(ETRI) 박사후연구원

1999~현재 제주대학교 초등컴퓨터교육전공 교수
관심분야: 컴퓨터교육
E-Mail: jkim0858@jejunu.ac.kr