

초등학생의 확산적 사고 촉진을 위한 CPS 프로그래밍 수업의 효과 분석

전성균[†] · 이영준^{††}

요 약

프로그래밍 교육은 문제를 가장 효율적으로 해결하기 위한 알고리즘을 찾아 구현하는 것이 일반적인 형태이다. 즉, 수렴적 사고 위주의 교육을 중시한다. 하지만 프로그래밍 교육은 그 대상에 따라 교육의 성격을 달리 해야 한다. 초등학교 학생들의 경우 효율성을 추구하는 것 보다 경험 중심의 다양한 탐색 환경이 선행되어야 한다. 이를 통하여 가장 효율적인 문제해결 방안을 스스로 찾아 가는 방법을 배워 나가야 할 것이다. 본 연구에서는 수렴적 사고의 비중이 큰 프로그래밍 교육을 초등학교 학습자 대상에 맞도록 확산적 사고와 수렴적 사고가 동시에 반복될 수 있는 확산적 사고 전략을 도입하여 그 효과를 분석하였다. 연구는 초등학교 5학년 학생들에게 적용되었으며, 일반적인 프로그래밍 수업을 실시한 통제집단과 CPS모형의 확산적 사고 전략을 포함한 프로그래밍 수업을 실시한 실험집단으로 구분하여 12차시의 실험처치 수업을 실시하였다. 그 결과 창의적 문제해결력의 하위요소 중 자기확신 및 독립성, 확산적 사고에 있어 CPS모형이 유의한 효과가 있는 것으로 나타났다.

주제어 : 창의성, 창의적 문제해결력, CPS, 스크래치, 프로그래밍

The Effect of CPS Programming Model on promoting Elementary school students' divergent thinking

Seongkyun Jeon[†] · YoungJun Lee^{††}

ABSTRACT

General form of the programming education is finding and realizing algorithm to solve problems faster and more efficiently. In other words, it is based on convergent thinking. However, the programming education must have different characteristics to education targets. For elementary school students, it is needed to provide various experience-centered investigation environments. They should learn how to find the most efficient problem solving method by themselves. This study had adopted divergent thinking strategy where divergent thinking and convergent thinking can be repeated at the same time to suit a programming education with great importance of convergent thinking to elementary school learners, and analyzed its effects. This study was applied to 5th graders, and 12 times of experimental measure classes were conducted by dividing them into the control group that conducted general programming class and the experimental group that conducted a programming class including divergent thinking of CPS model. As a result, CPS model had significant effect on the subordinate elements of creative problem solving skills, self-assurance, independence, and divergent thinking.

Keywords : Creativity, Creative Problem Solving Ability, CPS, Scratch, Programming

[†] 정 회 원: 용인 소현초등학교 교사

^{††} 중신회원: 한국교원대학교 컴퓨터교육과 교수(교신저자)

논문접수: 2011년 10월 09일, 심사완료: 2012년 03월 02일, 게재확정: 2012년 03월 23일

1. 서론

현대사회는 정보와 지식을 떼어놓고는 살아갈 수 없을 만큼 인간의 모든 부가가치가 정보와 지식으로 집중되어 있는 사회이다. 이러한 정보, 지식의 중요성으로 인하여 엄청난 속도로 지식의 양이 증가하고 있다. 이러한 시대를 살고 있는 우리들은 지식을 많이 알고 있는 것보다 넘쳐나는 지식들을 활용하여 문제를 해결하고, 지식의 형성과정과 구조를 이해하여 문제 상황에 적절히 변형, 적용하는 능력과 자기 주도적으로 지식을 재조직하여 독창적인 관점에서 문제를 해결하는 창의적 문제해결 능력이 강조되는 사회를 나아가고 있다[1].

컴퓨터 교육 분야에서도 이러한 시대적 변화를 반영하여, 지난 2005년 컴퓨터 과학에 대한 학습 내용을 강화하는 개정된 정보통신기술 교육 운영지침이 발표되었다. 개정된 정보통신기술 교육 운영지침의 궁극적인 목표는 학생들의 창의적 문제해결력의 신장이다[2]. 이에 따라 초등학교에서도 정보처리의 이해 영역에서 본격적으로 프로그래밍 교육이 도입되었다. 그러나 문법 위주의 교육과 정해진 프로그램을 따라하는 학습 과정 등 창의적 문제해결력 향상에 한계가 드러났지만, 그럼에도 불구하고 프로그래밍 교육을 통해 문제해결력 뿐만 아니라 창의성이 향상되었다는 다수의 연구 결과가 있고 이는 프로그래밍 교육을 통해 창의성을 향상시킬 수 있다는 가능성과, 적절한 교수학습 모형을 통해 그것이 가능하다는 의미를 지닌다고 볼 수 있다[3][4][5].

창의성은 일반적으로 유창성, 융통성, 독창성, 적절성, 정교성 등을 하위요소로 구분하고 있다. 유창성, 융통성, 독창성은 확산적 사고의 주요 구성 요소이며, 적절성과 정교성은 수렴적 사고의 주요 구성 요소로 구분된다[6].

프로그래밍 교육을 통하여 창의성이 향상되었다는 선행연구들은 창의성이라고 하는 구인이 향상되었다는 결과를 통하여 프로그래밍 교육이 효과가 있음을 주장하여 왔다.

배영권[3]은 창의적 문제해결력을 신장시키기 위해 로봇프로그래밍 교육 모형을 적용하였다. 그러나 실험 처치로 확산적 사고가 좀 더 활성화되어 창의적 문제해결력이 향상되었는지, 아니면 본래 프로그래밍 교육 특징으로 인해 수렴적 사고

가 활성화되어 결과적으로 창의적 문제해결력이 향상되었는지를 구분하여 분석할 필요가 있다.

창의성을 확산적 사고와 수렴적 사고로 구분하여 접근을 하게 되면 프로그래밍 교육의 접근, 창의성 측정, 구체적인 교수학습 전략의 개선점을 찾을 수 있다. 지금까지의 프로그래밍 교육은 최적의 해를 찾아가는 수렴적 사고를 큰 비중으로 두어 교육이 이루어져 왔으나, 이는 학습 대상별로 다르게 접근할 필요가 있다고 하는 것이 본 연구자의 주장이다. 프로그래밍 교육은 최적의 해를 찾아가는 수렴적 사고와 가능한 대안을 생각해보고, 오류를 검증 및 수정하는 과정에서 확산적 사고 과정이 포함되어 있다는 특징이 있다[7]. 하지만 초등학교 학습자의 경우 경험 중심의 다양한 탐색 환경이 선행되어야 하며, 이를 통하여 가장 효율적인 문제해결 방안을 스스로 찾아가는 방법을 배워 나가야 할 것이다. 창의적 문제해결력을 향상시키기 위해서는 문제해결 과정에 확산적 사고와 수렴적 사고를 동시에 반복적으로 경험하는 것이 필요하다[8].

본 연구에서는 수렴적 사고의 비중이 큰 프로그래밍 교육을 초등학교 학습자 대상에 맞도록 확산적 사고와 수렴적 사고가 동시에 반복되며 확산적 사고를 강화 할 수 있는 확산적 사고 전략을 도입하여 프로그래밍 수업을 설계하였다. 연구의 적용은 초등학교 5학년 학생들을 대상으로 선정하였으며, 일반적인 프로그래밍 수업을 실시한 통제집단과 CPS모형의 확산적 사고 전략을 포함한 프로그래밍 수업을 실시한 실험집단으로 구분하여 12차시의 실험처치 수업을 실시하여 그 효과를 분석하였다.

2. 이론적 배경

2.1 프로그래밍 교육

2.1.1 프로그래밍 교육에서의 확산적 사고

컴퓨터 프로그래밍은 '컴퓨터를 활용하여 학습자가 컴퓨터에게 자신이 원하는 것을 수행하도록 하는 작업'으로 정의된다. 이러한 컴퓨터 프로그래밍 교육은 프로그래밍 언어의 습득, 고등사고력 습득이라는 커다란 두 가지 목표를 지닌다. 언어의 습득은 명령어, 형식적 절차 및 변수 등과 같은 프로그래밍 일반에 대한 개념의 이해와 사용에 한정되며, 고등 사고력 향상

은 프로그래밍 과정에서 요구되는 문제 해결과 지속적 오류 검증, 수정 작업에서 요구되는 반성적 사고를 강조한다. 프로그래밍 교육이 이러한 목표를 가진다고 하였을 때 그 특징은 다음과 같이 정리할 수 있다.

첫째, 프로그래밍은 계획적이고 논리적인 사고활동이다. 둘째, 프로그래밍은 예측 가능하며 사고과정을 반추할 수 있는 활동이다. 셋째, 프로그래밍은 문제해결활동이며 결과 지향적인 활동이다. 넷째, 프로그래밍은 자기주도적 학습활동이며 창조적인 활동이다[7].

이와 같이 프로그래밍 과정 자체에 논리적 사고 과정, 문제해결 과정, 비판적 사고 과정을 포함하고 있다. 또한 오류 상황에서 디버깅을 하는 과정은 문제를 인식하고 문제해결을 위해 가능한 다양한 대안을 찾아보고 그 중 최적의 해결방안을 선택하는 사고 과정으로써, 확산적 사고와 수렴적 사고가 동시에 필요하다. 또한 다수의 연구에서 프로그래밍 교육이 학생들의 문제해결력 뿐만 아니라 창의성 신장에도 기여했다는 연구결과는 프로그래밍교육에 창의성에서 매우 큰 부분을 차지하고 있는 확산적 사고가 포함되어 있다는 점을 뒷받침해준다고 볼 수 있다[3][4][5].

2.1.2 교육용 프로그래밍 언어로서의 스크래치

일반적인 프로그래밍 교육의 학습의 효과는 Opera(1984)에 의하여 다음과 같이 제시되고 있다. 첫째, 컴퓨터 소양 능력을 키울 수 있다. 컴퓨터 소양은 컴퓨터의 능력이나 제한점뿐만 아니라 특정 언어의 명령을 학습하는 것까지 모두를 포함한다. 둘째, 수학적 성취와 긍정적 관련을 갖는 보다 수준 높은 인지과정을 의미하는 수학적 사고를 향상시킬 수 있다. 셋째, 프로그래밍 과정은 창조적 발명활동으로서 문제 해결과정 전반에 걸친 실험적 방법을 촉진하도록 유도하여 다양한 경험을 제공할 수 있다. 넷째, 프로그래밍 언어들은 논리의 기본적인 개념에 기초하고 있어 풍부한 논리적 추론과 언어의 고유의 논리적 원리를 배울 수 있다. 교육용 프로그래밍 언어란 교육적인 목적을 가지고 개발된 프로그래밍 언어를 지칭한다. 가장 대표적이고 오래 사용되어 왔던 것에는 Basic, Logo, Pascal 등이 있다. 이러한 언어들은 전 세계적으로 폭넓게 활용되었다.

교육용 프로그래밍 언어는 위에서 제시한 프로그래

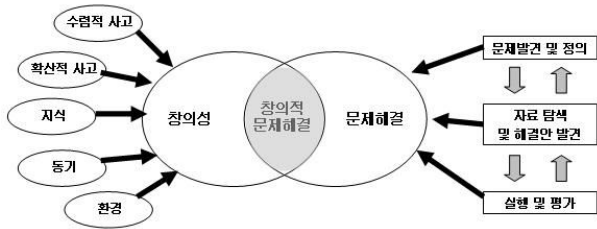
밍언어의 교육적인 효과들을 모두 포함하며, 주된 목적이 알고리즘적 사고력, 문제해결력을 키우는 것이므로 배우기 쉽고 다루기가 용이해야 한다[9][10]. 특히 최근 개발된 스크래치는 MIT공대, 미국과학재단, 인텔, UCLA 대학교에서 공동으로 개발한 새로운 교육용 프로그래밍 언어로서 학생들이 쉽게 애니메이션, 게임등을 만들 수 있게 해주는 ‘미디어 기반 프로그래밍 환경’이다[11]. 스크래치는 도구 자체 사용의 어려움으로 인한 인지부하를 감소, 실생활의 문제해결과정과 유사한 경험의 제공, 시각적이고 상호작용적인 프로그래밍 환경을 제공, 즉각적이고 구체적인 피드백을 제공 등의 특징을 지닌다. 이러한 스크래치의 환경적 특징은 어떤 현상에 대한 문제를 발견하고 오류를 수정하는 데에 필요한 사고 능력 향상에 도움을 줄 수 있는 긍정적인 요인으로 작용한다[12].

2.2 창의적 문제해결력

Guilford(1959)는 인간의 사고를 수렴적 사고와 확산적 사고로 나누고, 그 구성요인으로 사고의 속도를 나타내는 유창성, 사고의 넓이를 나타내는 융통성, 사고의 참신성을 나타내는 독창성, 남들이 보지 못하는 부분을 파악하는 정교성, 사물에 대해 반응하는 민감성, 재정의 및 재구성력 등을 제시하였다[6]. Arieti(1976)는 창의적사고의 지적 요인으로 기억, 인지, 평가, 수렴적 사고, 확산적 사고를 들고 있다[13]. Firestein과 Treffinger(1983)는 창의성의 구성요인으로 수렴적 사고의 중요성을 강조하며 신기성, 가치성, 전환성, 농축성을 제시하였다[14].

창의적 문제해결은 해결의 통로가 하나가 아니라 여러 개이며, 이 여러 해결 대안들 가운데서 최선의 대안을 찾아가는 과정이다[15]. 또한, 창의적 문제해결이란 창의성이 발현되는 과정 또는 여러 수행들 중 일부이며, 동시에 문제해결 중에서 창의성이 요구되는 특별한 형태로서 <그림 1>와 같이 표현될 수 있다. 이 그림에서 창의적 문제해결은 창의성의 속성과 문제해결의 속성을 모두 만족시켜야 하는 영역이며, 두 관점을 모두 수용한 포괄적인 입장을 나타내고 있다[16].

이상에서와 같이 창의성, 문제해결, 문제해결과 창의적인 사고과정을 연결 짓는 연구들을 종합하여 한국교육 개발원(2001)에서는 창의적 문제해결



<그림 1> 창의성, 문제해결, 창의적 문제해결의 관계

력을 “문제 해결과정에서 다양한 요인이 복잡적이며 역동적으로 상호작용 하여 문제 해결에 유용하며 독창적인 산출물 또는 해결책을 만들어내는 것”으로 정의하였다[17].

2.3 CPS 모형(Creative Problem Solving : 창의적 문제해결 모형)

Parnes는 CPS모형을 1970년대와 1980년대 교육현장이나 기업 등에 광범위하게 보급하였다[18]. 그 후, Isaksen과 Treffinger(1987)에 의해 3요소와 6단계로 순환구조의 CPS모형이 만들어졌다. 여기서 ‘요소’라 함은 문제해결을 창의적으로 하려고 할 때 사람들이 관여하는 활동의 범주 또는 일반적 영역이라 말할 수 있다. 세 가지 요소는 문제의 이해, 아이디어 생성 및 행위를 위한 계획 등이다. 요소 보다 더 구체적인 수준에 단계가 있다. 문제의 이해 속에 관심영역 발견, 자료 발견, 문제 발견 단계들이 포함된다. 아이디어 생성에는 아이디어 발견 단계가 있다. 그리고 행위를 위한 계획에는 해결발견과 수용발견의 두 가지 단계들이 있다. 그리고 각 단계에는 확산적사고와 수렴적 사고의 두 가지 국면이 있어 이들 간의 역동적 균형을 강조하고 있다[15][19].

본 연구에서는 Parnes가 제시한 5단계 모형을 적용하였다. Parnes의 모형은 교육현장에 실제 적용하기 위해 만들어졌고 광범위하게 활용되었다. 그리고 각 단계를 고정적으로 적용하는 것이 아니라 필요에 따라 융통성 있게 적용할 수 있다는 점이 CPS의 중요한 특징 중의 하나이다. 또한 실제 단위 수업에 적용하기 위해서는 선형모형이 보다 현실적인 대안이 될 수 있다.

창의적 문제해결모형은 <그림 2>와 같이 각각의 단계에 확산적 사고와 수렴적 사고의 반복을



<그림 2> Parnes의 CPS모형

거쳐 창의적으로 문제를 해결하도록 한다. 본 연구에서는 확산적 사고와 수렴적 사고를 균형 있게 향상시킬 수 있는 CPS 모형을 프로그래밍 수업 설계의 기본 모형으로 선택하였고 다양한 확산적 사고 지도전략을 활용하여 학생들의 탐색 활동을 극대화하고자 하였다.

2.4 확산적 사고지도 전략

학생들의 창의성이나 창의적 문제 해결력을 신장시키기 위해서는 보다 직접적이고 적극적인 교수학습 전략을 세워 지도할 필요가 있다. 본 연구에서는 전통적인 창의성 기법 중 확산적 사고를 향상시키기 위한 창의성 전략을 교수학습 과정에 포함시켜, 프로그래밍 과정 동안 학생들이 문제해결 전략에 접근하기 위하여 다양한 사고를 할 수 있는 기회를 제공하였다. 확산적 사고를 위하여 본 연구에서 활용된 창의성 기법은 브레인스토밍, 스캐퍼, 형태분석법, PMI이다.

브레인스토밍이란 ‘뇌에 폭풍을 일으킨다’는 의미로 두뇌의 능력을 최대한 발휘하여 아이디어를 이끌어내는 것을 말한다. 여러 개의 대안이 필요할 때, 다양한 가능성을 고려해 보고 싶을 때, 다양한 시각에서 아이디어들을 생성해 보고 싶을 때 사용한다[20]. 본 연구에서는 문제발견, 아이디어 생성과정에 있어 블록의 활용과 학습 활동의 실행과정에서 본 기법을 활용하였다.

스캐퍼는 일련의 주어진 정보의 변형에 대한 7가지 사고 방법들의 약어로서 특정대상이나 특정문제 대하여 대체(Substitute), 결합(Combine), 적용(Adapt), 수정, 확대/축소(Modify-Magnify/Minify), 다른 용도(Put to other use), 제거(Eliminate), 재배열/거꾸로(Rearrange/Reverse)를 필요에 따라 선택해 새로운 아이디어를 찾는 방법[21]으로 본

연구에서는 문제 발견, 해결 발견 단계에서 다양한 정보 변형을 통하여 문제에 접근하도록 유도하는데 활용되었다.

형태분석법(morphological analysis)은 문제 상황에서 대상의 요인을 찾아내고, 각 요인의 속성을 열거하여 도표를 만들어 새로운 아이디어를 각 칸에 기록해 보도록 하는 방법[16]으로 스캐퍼와 마찬가지로 해결 발견 단계에서 주로 활용되었으며, 스프라이트의 위치와 종류 등을 다양화하는데 매우 효과적으로 활용되었다.

PMI의 아이디어 처리는 세 가지로 나뉘는데, 각각의 영어 첫 철자를 따서 제안되었다. 좋은 점(Plus), 나쁜 점(Minus), 흥미로운 점(Interest) 등의 세 가지 관점에서 생각하여 새로운 아이디어를 찾는 방법이다 [22]. PMI는 주로 평가활동에 확산적 사고를 위한 기법으로 활용되었으며, 다른 학생의 작품 및 산출물과 자신의 것과의 비교, 장단점 분석 등에 활용되었다.

3. 확산적 사고 전략을 포함한 CPS프로그래밍 수업 설계

3.1 수업 구조화 전략

프로그래밍과 다양한 창의성 기법들을 처음 접해보는 학생들의 인지 부담을 최소화하기 위해서 학습 내용을 3단계로 구조화하여 제시한다.

1단계인 ‘프로그래밍 기초 개념 익히기’에서는 교육용 프로그래밍 언어 스크래치를 소개하고 이를 이용하여 기초 개념 및 기초 기능을 학습할 수 있도록 하였다.

2단계인 ‘간단한 프로그래밍 학습하기’에서는 스크래치를 통해 만들 수 있는 프로그램을 본격적으로 학습하는 단계로써 본격적으로 확산적 사고를 촉진하기 위한 다양한 전략을 사용한다.

3단계인 ‘다양한 프로그래밍 학습하기’에서는 다양한 프로그램을 작성할 수 있는 교육 상황을 제시해야 하며 점차적으로 개방형 문제를 제시함으로써 창의적 문제해결력이 향상될 수 있도록 한다.

3.2 수업 내용 구성

앞서 살펴본 수업 구조화 전략에 따라 학생들에게 적용할 프로그래밍 수업의 내용 구성은 <표 1> 과 같다.

<표 2>는 교수 학습 과정의 예시를 제시하였다.

<표 1> CPS프로그래밍 수업의 내용 구성

차시	학습 주제	주요 내용 및 활동
1-3	프로그래밍 기본 개념 익히기	스크래치 프로그래밍 언어 소개 및 기본적인 프로그래밍 학습 - 스크래치 특징 및 화면 구성 - 블록을 이용한 프로그래밍
4-7	간단한 프로그래밍 학습하기	스크래치를 이용하여 프로그래밍 하기 - 조건문 활용 - 제어문 활용 - 반복문 활용
8-12	다양한 프로그램 만들기	스크래치를 이용하여 여러 가지 재밌는 프로그램을 만들어 보기 - 재밌는 프로그램 만들기 - 전자책 만들기

<표 2> 6차시 교수 학습 과정의 예

학습 목표		조건문과 제어문을 이용하여 프로그래밍 할 수 있다.							
단계	학습	활동 내용		확산적 사고 전략					
사실 발견	동기 유발	▶ 동기유발: 엘리베이터 1층을 눌렀을 때 각각의 상황을 생각해보며 동작원리를 찾아본다. ▶ 학습목표: 조건문과 제어문을 이용하여 프로그래밍 할 수 있다.		다양한 상황을 생각할 수 있도록 점진적으로 발문한다.					
	문제 확인	▶ 움직이며 중얼거리는 고양이 프로그램을 보며 동작원리를 찾아본다 ▶ 분석한 프로그램을 실제로 구현한다.							
문제 발견	활동1	▶ 조건문과 제어문을 활용하여 스프라이트가 왼쪽에 있으면 삼각형을, 오른쪽에 있으면 원을 그리는 프로그램을 만들기 위해 브레인스토밍으로 활용 가능한 블록들을 찾아본다.		브레인스토밍으로 삼각형 및 원을 그리는 방법을 자유롭게 적어본다.					
아이디어 발견	활동2	▶ 형태분석법을 이용하여 새로운 프로그램을 만들어본다.							
	해결 발견	활동3	<table border="1" style="display: inline-table; margin-right: 10px;"> <tr> <th>스프라이트 종류</th> <th>좌측에 위치할 때 효과</th> <th>우측에 위치할 때 효과</th> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </table>	스프라이트 종류	좌측에 위치할 때 효과	우측에 위치할 때 효과			
스프라이트 종류	좌측에 위치할 때 효과	우측에 위치할 때 효과							
수용 발견	정리	▶ 자신의 프로그램과 온라인 공유된 다른 친구의 프로그램을 PMI로 평가한다.		자신과 친구의 프로그램을 비교 분석하며 장단점을 비교해본다.					
		<table border="1" style="display: inline-table; margin-right: 10px;"> <tr> <th>친구 이름</th> <th> </th> <th> </th> </tr> <tr> <td>더하기(Plus)</td> <td>빼기(Minus)</td> <td>흥미로운점(Interesting)</td> </tr> </table>	친구 이름			더하기(Plus)	빼기(Minus)	흥미로운점(Interesting)	
친구 이름									
더하기(Plus)	빼기(Minus)	흥미로운점(Interesting)							

3.3 CPS 단계별 지도 방안

확산적 사고 촉진을 위하여 창의성 기법 들이 CPS모형 전반에 걸쳐 활용될 수 있도록 하였으며, 프로그래밍 교육과 확산적 사고 촉진 전략의 효과성을 극대화하기 위하여 각 단계별 특징과 프로그래밍 교육의 특성을 반영하여 전체 프로그램의 전개과정을 도출하였다.

3.3.1 사실발견 단계

문제를 파악하기 위한 정보와 자료를 수집하고 분석하는 단계로써 수업의 도입부라고 할 수 있다. 따라서 이 단계에서는 정보를 수집하여 그 상황에서 무엇이 진행되는지 학생 스스로 찾아 볼 수 있도록 발문하는 것이 중요하다. 호기심과 동기를 유발할 수

있도록 사전에 조직된 개방된 질문을 제시하였다.

3.3.2 문제발견 단계

다양한 관점으로부터 가능한 문제를 탐색하여 문제를 정의하고 인식하는 단계이다. 전 단계에서 탐색한 사실을 바탕으로 문제가 무엇인지 확산적 사고 과정에서 다양하게 문제를 진술해보고 수렴적 사고 과정을 통해 최선의 문제 진술을 선택하도록 해야 한다. 따라서 브레인스토밍 기법을 활용하여 가능한 대안을 생성해보는 활동을 주요 활동으로 선정하였다.

3.3.3 아이디어발견 단계

문제발견 단계에서 최종적으로 진술된 문제를 해결할 아이디어를 생각해 내는 단계이다. 이 단계에서는 확산적 사고에 보다 초점을 두어야 하고 따라서 다양한 확산적 사고 촉진 전략을 사용하여 아이디어를 많이 생성해 내도록 하였다.

3.3.4 해결발견 단계

산출된 여러 가지 아이디어들을 어떤 기준에 근거하여 평가하고 최선의 아이디어를 선정하는 단계이다. 이 단계에서는 스크래치를 이용하여 프로그래밍을 하기 위해서 실제로 구현 가능한지도 평가 기준에 포함되어야 한다. 이 단계에서는스캐퍼, 형태분석법, PMI등 다양한 전략을 통하여 해결 방안을 도출해내는 활동을 하였다.

3.3.5 수용발견 단계

선택된 아이디어를 바탕으로 실천에 옮길 행동 계획을 세우는 단계이다. 즉 실제로 스크래치로 프로그래밍 하여 결과를 확인하는 단계이다. 실제로 프로그래밍 수행 결과 예상치 못한 오류가 나타났을 경우에 학생 스스로 오류를 해결하기 위해 다양한 확산적 사고 촉진 전략을 선택하여 활용하는 모습을 확인할 수 있었으며, 다른 학생들의 산출물과 본인의 산출물을 비교, 장단점 분석을 하는데 PMI 기법을 주로 활용하였다.

4. 연구 방법

확산적 사고 중심의 프로그래밍 교육이 학생들의 창의적 문제해결에 미치는 영향을 살펴보기

위해 용인에 소재한 S초등학교 5학년 학생들을 연구 대상으로 선정하였다. 발달단계상 형식적 조작기에 진입하는 단계로 본 연구에서 적용하는 프로그래밍이라는 추상적 사고 과정을 다양한 전략 및 구성으로 학습할 수 있는 대상이라 판단된다. 연구 대상자는 실험집단 1학년 33명, 통제집단 1학년 31명으로 구성하였다.

4.1 실험설계 및 절차

실험처치 전 사전검사를 실시하고 두 집단의 동질성 여부를 분석하였다. 이후, 실험집단과 통제 집단 모두 스크래치 기본 사용 방법 및 기초적인 프로그래밍 교육을 3차시에 걸쳐 실시하였으며, 이어서 실험집단은 CPS모형의 확산적 사고 단계 중심의 프로그래밍 수업 12차시를, 통제집단은 일반적인 방식의 프로그래밍 수업 12차시 실시하였다. 일반적인 방식의 프로그래밍 수업은 확산적 사고를 촉진하기 위한 전략들을 배제하고 단지 학습자에게 학습 내용을 전달하고 실습할 수 있는 환경을 제공하는 형태로 진행하였다. 이러한 실험 설계를 정리하면 <표 3>과 같다.

<표 3> 실험설계

G1	O1	X1	O2
G2	O3	X2	O4

- G1 : 실험집단
- G2 : 통제집단
- O1 O3 : 사전검사(창의적 문제해결력 검사)
- O2 O4 : 사후검사(창의적 문제해결력 검사)
- X1 : 확산적 사고 전략을 활용한 CPS프로그래밍 수업
- X2 : 기존의 프로그래밍 수업

4.2 연구 도구

본 연구의 사전·사후 검사에서 사용하는 창의적 문제해결력 검사 도구는 한국교육개발원에서 발간한 ‘간편 창의적 문제해결력 검사 개발 연구(I)’(2001)를 기반으로 2004년 서울대 심리 연구실 MI 연구팀에서 개발하였다. 개발된 도구는 정의적 영역의 자기 확신 및 독립성, 동기적 사고, 인지적 영역의 확산적 사고, 비판적·논리적 사고, 영역으로 구성되어 있다. 각 영역별 5문항으로 각 문항의 점수는 5점 척도로 구성되어 있고 신뢰도는 Cronbach α 값이 .89로 양호하게 나타났다.

5. 연구 결과

5.1 사전검사 결과

창의적 문제해결력 사전검사 결과, <표 4>와 같이 실험집단과 통제집단은 통계적으로 유의한 차이가 없는 것으로 나타났다. 즉 두 집단이 동질 집단임을 확인하였다.($p>.05$)

<표 4> 창의적 문제해결력에 대한 사전검사 결과

영역	집단	사례수	평균	표준편차	t	p
자기확신 및 독립성	실험	33	15.33	3.323	1.086	.282
	통제	31	14.26	4.538		
확산적 사고	실험	33	15.94	3.211	.158	.875
	통제	31	15.81	3.525		
비판적·논리적 사고	실험	33	17.45	3.483	-1.250	.216
	통제	31	18.55	3.731		
동기적 요소	실험	33	18.42	3.072	-1.014	.315
	통제	31	19.16	3.012		
전 체 (창의적 문제해결력)	실험	33	67.15	11.503	-.232	.817
	통제	31	67.77	12.224		

5.2 사후검사 결과

창의적 문제해결력의 하위요소별로 실험집단과 통제집단 간의 차이를 살펴본 독립표본 t-검정한 결과 <표 5>와 같이, 두 집단은 창의적 문제해결력 각 하위 요소인 자기 확신 및 독립성, 확산적 사고는 통계적으로 유의한 차이가 있는 것으로 나타났다($p<.05$). 반면 비판적 논리적 사고 및 동기적 요소는 유의한 차이가 없는 것으로 나타났다 ($p>.05$). 그러나 전체 창의적 문제해결력은 유의한 차이를 보였다($p<.05$)

이는 실험처치로 확산적 사고 촉진 전략을 적용하여 학생들이 수업 중에 다양한 아이디어를 이끌어 내는데 긍정적인 영향을 주었을 것이라 분석하였다. 반면 프로그래밍 교육을 처음 접해보는 두 집단 모두 수업 시간에 큰 관심을 보이며 직접 원하는 프로그램을 만들어 실행하는 등 학습에 몰입하여 동기적 요소에서는 유의한 차이가 발생하지 않은 것으로 나타났다. 또한 비판적, 논리적 사고 영역 역시 두 집단이 유의한 차이가 나타나지 않았다. 프로그래밍 교육이 학생들의 논리적 사고 및 비판적 사고력에 통제 집단, 실험집단에 긍정적 영향을 미쳤고, 또한 두 집단간에 의미있는 차이는 나타나지 않은 것을 통해 프로그래밍 수업이 충실히 진행되었고, 또한 본 연구에서 적용한 확산적 사고 전략이 기존의 프로그래밍

교육에 유기적으로 통합되었다고 분석된다.

<표 5> 창의적 문제해결력에 대한 사후검사 결과

영역	집단	사례수	평균	표준편차	t	p
자기확신 및 독립성	실험	33	16.52	2.774	2.617	.011
	통제	31	14.35	3.782		
확산적 사고	실험	33	17.55	3.173	2.643	.010
	통제	31	15.26	3.741		
비판적·논리적 사고	실험	33	19.03	2.672	.254	.800
	통제	31	19.10	2.868		
동기적 요소	실험	33	20.21	2.414	.896	.374
	통제	31	19.61	3.422		
전 체 (창의적 문제해결력)	실험	33	73.30	8.902	2.126	.038
	통제	31	68.32	10.934		

6. 결 론

일반적으로 프로그래밍 교육은 문제를 가장 빠르고 효율적으로 해결하기 위한 알고리즘을 찾아 구현하는 것을 목표로 한다. 즉, 수렴적 사고 위주의 교육을 중시한다. 하지만 프로그래밍 교육을 처음 접하거나 경험이 부족한 초등 학습자에게도 동일한 학습 목표를 가지고 효율성을 강조하여야 할 것인가는 재고해 봐야 한다. 프로그래밍 교육을 포함한 모든 교육은 학습자의 상황과 발달단계를 고려하여야 한다. 초등 학습자들에게는 효율성을 추구하는 것 보다 경험 중심의 다양한 탐색 환경을 경험할 수 있도록 해야 한다. 초등 학습자들은 정해진 해법의 문제를 정확하고 빠르게 누군가 알려주는 것을 원하지 않는다. 물론 누구의 산출물이 어떠한 이유로 인해 더욱 효율적이고, 그 프로그램의 장단점이 무엇인지 분석하는 것은 필요하다. 하지만 그 결과에 이르기까지의 과정을 다양하게 만드는 확산적 사고는 초등학습자의 학습 발달 단계에서 매우 중요한 요소일 것이다.

본 연구에서는 수렴적 사고의 비중이 큰 프로그래밍 교육을 초등학교 학습자 대상에 맞도록 확산적 사고와 수렴적 사고가 동시에 반복될 수 있는 확산적 사고 전략을 도입하여 프로그램을 설계하여 적용하였다.

연구 결과 학생들의 창의적 문제해결력에 긍정적인 영향을 주었고 이는 정형화된 답 이외에도 다양한 아이디어를 생성해내도록 확산적 사고를 촉진하는 다양한 전략을 학생들에게 적용함으로써 창의적 문제해결력 전체에 긍정적인 영향을 미쳤다고 볼 수 있다. 즉 CPS모형을 기반으로 확산적

사고와 수렴적 사고 교육이 동시에 이루어질 수 있도록 확산적 사고를 촉진하는 전략을 더함으로써 내실 있는 프로그래밍 교육이 이루어졌고 본 연구의 결과는 향후 초등학교의 프로그래밍 교육의 방향 설정에 있어 큰 도움이 될 것이다.

참 고 문 헌

[1] 이영준, 임웅, 이은경 (2010). 창의적 문제해결력 향상을 위한 정보 교육 프로그램. **컴퓨터교육학회논문지**, 13(1), 1-8.

[2] 교육인적자원부 (2005). 초·중등학교 정보통신기술교육 운영 지침

[3] 배영권 (2006). **창의적 문제해결력 신장을 위한 유비쿼터스 환경의 로봇 프로그래밍 교육모형**. 박사학위 논문, 한국교원대학교.

[4] 이민희 (2009). **두리틀을 이용한 프로그래밍 수업이 창의성, 문제해결력, 프로그래밍 흥미도 향상에 미치는 영향**. 석사학위 논문, 전주교육대학교 .

[5] 정은영 (2008). **Squeak Etoys 기반 정보교육이 초등학생의 창의적 문제해결력에 미치는 영향**. 석사학위논문, 한국교원대학교.

[6] Guilford, J. P (1959). *Creativity and it's cultivation*. New York: Harper and Row.

[7] 이인지 (2004). **정보 컴퓨터**. 서울: 도서출판 희소.

[8] 김영채 (2007). **창의력의 이론과 개발**. 서울: 교육과학사.

[9] Papert, S. (1980). *Mindstorms: children, computers, and powerful idea*. New York: Basic Books.

[10] 이옥화 (1993). 로고 프로그래밍의 교육적 의의와 실천 방안 모색. **교육공학 연구**, 8(1), 81-102.

[11] J. Maloney, L. Burd, Y. Kafai, N. Rusk, B. Silverman, and M. Resnick (2004). Scratch: A Sneak Preview. In *Second International Conference Creating, Connecting, and Collaborating through Computing*, pages 104-109, Kyoto, Japan.

[12] 이은경 (2009). 문제해결력 향상을 위한 과제 중심 스크래치 프로그래밍 학습 프로그램. **컴퓨터교육학회논문지**, 12(6), 1-9.

[13] Arieti, S. (1976). *Creativity: The magic synthesis*. NY: Basic Books.

[14] Firestein, R. L., & Treffinger, D. J (1983). Ownership and converging: essential ingredients of creative problem solving. *Journal of Creative Behavior*, 17, 32-38.

[15] 김영채 (1999). **창의적 문제해결: 창의력의 이론, 개발과 수업**. 서울: 교육과학사

[16] 조연순, 성진숙, 이해주 (2008). **창의성 교육**. 서울: 이화여자대학교 출판부. 192-194, 198.

[17] 한국교육개발원 (2001). **간편 창의 문제해결력 검사 개발연구(I)**.

[18] Parnes, S. J. (1967). *Creative behavior guidbook*. NY: Charles Scribner's Sons.

[19] Isaksen, S. G., Dorval, K.b., & Treffinger, D. J. (2000). *Creative approaches to problem solving: A framework for change*. Dubuque, Iowa: Kendall/Hunt.

[20] 임규혁, 임웅 (2008). **교육심리학**. 서울: 학지사. 105.

[21] 주상윤 (2007). **창의성 발상의 원리와 기법**. 울산: 울산대학교출판부. 167-172.

[22] 한순미, 김선, 박숙희, 이경화, 성은현 (2005). **창의성**. 서울: 학지사. 192-197.



전 성 군

2003 한국교원대학교
초등교육과(교육학석사)
2011 한국교원대학교
컴퓨터교육과 (교육학석사)

현재 용인 소현초등학교 교사
관심분야 : 정보교육, 로봇프로그래밍, 학습과학
E-Mail : presents@empas.com



이 영 준

1988 고려대학교 전산과학과
(이학사)
1994 미국 미네소타대학교
(전산학 Ph.D.)

현재 한국교원대학교 컴퓨터교육과 교수
관심분야 : 정보통신교육, 지능형시스템, 학습과학
E-Mail : yjlee@knue.ac.kr