

초등학생들의 창의력과 논리력 향상을 위한 프로그래밍 언어 교수전략에 관한 연구

김 갑 수

서울교육대학교 컴퓨터교육과

요 약

초등학교 교육에서는 창의성과 논리력 향상을 강조하고 있고, 이를 위한 많은 연구가 진행되었지만, 컴퓨터 프로그래밍 교육에서 교수 전략이 매우 부족하다. 이에 본 연구에서는 창의력과 논리력 향상을 위한 프로그래밍 언어 교수 전략을 개발한다. 본 연구에서 개발한 교수 전략은 입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리, 추상화의 원리를 만들었다. 이 5개의 원리들을 초등학교 현장 전문가 20명에게 심층 면접으로 검증하였다. 초등학교 교사들이 본 연구의 결과를 이용하면 프로그래밍 언어 교육에 많이 기여할 것이다.

키워드 : 프로그래밍 언어 교육, 창의성, 논리성.

A Study on Programming Language Instruction Strategies of Improving the creative and logical thinking for Elementary Students

Kapsu Kim

Seoul National University of Education, Department of Computer Education

ABSTRACT

Logical thinking and creative thinking in the elementary school are very important. Even though these have been studied very many, the computer program education has the lack of instruction strategy. In this paper, the instruction strategies of computer program language education in the elementary school to improve them are proposed. These are the principles of the input and output, effectiveness, generalization, simplification, and the principles of abstraction. These five principles are verified by the 20 experts from deep interview. Using the results of this study will contribute a lot of the program language education for elementary school teachers.

Keyword : programming language education, creative thinking ,logical thinking.

본 논문은 2007년도 학술진흥재단 기초연구 과제 지원(327-2007-2-B00636)에 의해 지원되었음.

논문투고 : 2009-01-24

논문심사 : 2010-02-10

심사완료 : 2010-02-17

1. 서 론

컴퓨터 교육은 세계적인 추세로서, 미국과 유럽 등에서는 컴퓨터의 기본 교육 및 프로그래밍 교육에 대한 교육과정 등을 만들어서 교육하고 있다[12], 우리나라도 2001년부터 ICT(Information and Communication Technology) 교육 지침에 의해 초등학교부터 컴퓨터 교육을 실시하고 있다. 2005년 12월에 개정된 정보통신기술 교육에서 “창의력, 문제해결력, 논리적 사고력 등 고등 사고 능력을 함양하여 미래 지식 정보 사회를 유지·발전시킬 수 있는 잠재적 인재를 육성”해야 한다고 강조하였다. 그 내용체계의 5단계 중 ‘정보처리의 이해’ 영역에 ‘프로그래밍의 이해와 기초’라는 소영역으로 반영되어[2] 초등학교 5학년과 6학년 학생들에게 프로그래밍 언어 교육을 강조하고 있다.

Hijazi, Sam, Smith, M. Leigh 와 Alvarado-Vazquez, Edgardo는 창의적인 사고를 위해서 프로그래밍 교육의 중요성을 설명하였다[7]. Van Lengen과 Craig A는 프로그래밍 경험을 통하여 추상적인 개념을 이해할 수 있다고 하였다[3]. Salomon G와 Perkins D.N은 컴퓨터 프로그래밍 학습을 통하여 지식의 전이와 문제 해결에 도움이 된다[11]고 주장한 내용으로 보아 프로그래밍 언어 교육이 창의적인 사고와 논리적이 사고 형성에 도구로 사용된다는 것을 알 수 있다.

최근 놀이 활동이나 LOGO 언어, 로봇을 활용한 프로그래밍 교육에 관심이 높아지고 있다[1]. 그러나 부분적으로 프로그래밍 언어 학습을 위한 교수 방법의 탐색[4]에 대한 연구가 진행되었지만 프로그래밍 언어 교육을 위한 교수 전략 등을 개발하지 못하고 있다. 본 연구에서는 프로그래밍 언어가 창의성과 논리성 계발을 위하여 필요하기 때문에 초등학교생들에게 쉽게 프로그래밍 언어 학습을 위한 프로그래밍 언어의 특성을 반영한 프로그래밍 언어 교육의 전략을 개발할 필요 있다.

본 논문의 2장에서는 기존 프로그래밍 교육에 관한 연구들을 간단히 분석하고, 3장에서는 논리성과 창의성 계발을 위한 프로그래밍 언어 교육 교수 전략들을 설명하고, 4장에서는 제안한 프로그래밍 언

어 교수 전략에 대한 전문가 집단의 검증 결과를 설명하고, 5장은 본 연구의 결론이다.

2. 관련 연구

2.1 프로그래밍 교육과 인지 능력

프로그래밍 교육과 인지 능력에 관한 연구는 Salomon G와 Perkins의 연구와 Dona M. Kagan의 연구가 있다. 먼저, Salomon G와 Perkins의 연구를 살펴본다.

Salomon G와 Perkins[11]는 컴퓨터 프로그래밍 언어 교육을 함으로서 학생들의 인지 기술이 전이 된다고 하였다. 이 연구자들은 컴퓨터 프로그래밍 교육을 함으로서 인지 능력이 6개의 범주로 나누어 전이 된다고 하였다. 6개의 범주에 대한 상세한 설명은 다음과 같다.

첫 번째 범주는 프로그래밍 언어 교육을 함으로써 수학적 기하학적 개념과 원리를 향상할 수 있다는 것이다. 프로그래밍 언어는 수학의 도형에 관련된 라이브러리가 있어서 이를 이용하여 바로 프로그래밍을 하면 수학적 기하학적 개념과 원리가 향상할 수 있다는 것이다. 또한 변수, 수식 및 함수의 표현 등을 원하는 형태대로 변경할 수 있기 때문에 수학적 기하학적인 지식의 전이가 일어난다는 것이다.

두 번째 범주는 프로그래밍 언어 교육을 함으로써 문제 해결, 문제 찾는 능력과 문제 관리 능력이 생긴다는 것이다. 이것은 학생들이 컴퓨터 프로그래밍을 하면서 문제를 해결하는 과정에서 에러를 찾고 이를 수정하면서 구체적인 문제 해결과정을 관리할 수 있는 능력이 생긴다는 것이다.

세 번째 범주는 프로그래밍 언어 교육을 함으로써 논리적인 추론과 표현 능력을 향상할 수 있다는 것이다. 컴퓨터 프로그래밍을 하면서 어떤 상황을 형식적으로 표현하지 않으면 프로그래밍을 할 수 없다. 어떤 사항을 형식적인 논리로 구성해서 논리적 수학적 모델링을 할 수 있으면 프로그래밍을 할 수 있다.

네 번째 범주는 프로그래밍 언어 교육을 함으로써 지식, 생각, 학습의 모델링 능력을 향상할 수 있

다는 것이다. 프로그래밍 하는 작업은 정보를 가공하여 처리하는 능력을 갖게 하는 것이다. 정보를 가공하면 지식이 되고 정보의 가공 능력 향상은 지식, 생각 및 학습의 향상을 기할 수 있다.

다섯 번째 범주는 프로그래밍 언어 교육을 함으로써 인지 유형을 만들 수 있다는 것이다. 프로그래밍은 어떤 문제를 해결하기 위해서 프로그래밍 작성자가 실제 구현하는 것이다. 프로그래밍을 하는 것은 학생들이 경험하면 자신의 인지 유형을 만들 수 있다.

여섯 번째 범주는 프로그래밍 언어 교육을 함으로써 열정과 인내를 가질 수 있다는 것이다. 프로그래밍을 할 때에는 열정과 인내가 필요하다. 톰슨이 UNIX의 기본 커널을 만들 때에 한 달 만에 만들었다고 한다. 이처럼 열정과 인내를 가지고 만들어야 프로그래밍이 완성된다.

Salomon G와 Perkins 연구는 프로그래밍 언어 교육을 위한 교수 전략을 제안하지 않았다. 본 연구에서 프로그래밍 언어의 교수 전략을 개발하여 Salomon G와 Perkins의 6개의 지식 전이를 촉진하게 할 수 있게 한다.

다음은 Dona M. Kagan의 연구를 살펴본다. Dona M. Kagan[6]는 교사가 교실에서 학생들을 교수할 때에 지식 구조와 인지과정에서의 교수 설계가 중요하다고 하였다. 실제 교수 설계자는 컴퓨터 프로그래밍의 인지 연구로부터 많은 이론적인 기반을 구축하여야 한다고 했다.

Dona M. Kagan은 지식 구조와 인지과정에서의 교수 설계가 중요성을 지적하였지만 이를 구체적인 방법론은 제시하지 않았다.

2.2 프로그래밍 교육과 교수 기법

프로그래밍 교육과 교수 기법에 대한 연구는 Angela Carbone, Linda Mannila, 및 Sue Fitzgerald의 연구와 Jeroen Keppen과 David Hay의 연구가 있다.

먼저, Angela Carbone, Linda Mannila, 및 Sue Fitzgerald를 살펴본다. Angela Carbone, Linda Mannila, 및 Sue Fitzgerald(2007)[5]는 정보 기술 교육을 어떻게 가르치면 성공적으로 가르치게 되는

것이고 어떤 방법으로 가르치면 성공적이지 않는지에 대한 연구를 하였다. 정보 기술을 담당하는 교사들을 심층적으로 조사하여 정보 기술을 성공적으로 가르치지 않는 경우에는 교수 기술이 부족하고, 조직적인 지원이 부족하고, 학생들에게 책임감을 주지 못하고, 복잡한 문제들에 대한 학생들이 이해하지 못한 것이다. 강의시간이 많고, 경험 부족, 관리 부족 및 흥미 부족 등도 프로그래밍 교육을 성공적으로 할 수 없다고 했다.

이를 보면 정보 기술을 강의하는 교사는 성공적인 교사가 되기 위해서 교수 기법에 대한 철학을 가져야 한다는 것이다. 본 연구에서는 교사들이 프로그래밍 언어를 교수하기 위한 방법으로 내용적인 측면에서 교수 기술이 부족한 부분을 만들 수 있고, 본 연구에서 제안한 프로그래밍 교수 전략이 이 부분을 해소할 수 있다.

다음은 Jeroen Keppen과 David Hay의 연구를 살펴본다. Jeroen Keppen과 David Hay[8]은 효율적인 교수 방법은 “학생들이 접하는 재료가 친근감이 있어야 하고, 학생들이 관심을 갖고 있는 정도를 측정하고 감독하는 것이다.”라고 하였다. 그래서 개념 지도(Concept mapping)를 사용하여 학생들이 학습하고자 하는 내용을 모델링할 수 있어야 하고 학생들이 만들고자 하는 것을 개념지도와 같은 그래프를 표현할 수 있어야 한다. 따라서 컴퓨터 프로그래밍을 지도할 때에 개념을 그래프로 표현하는 능력이 필요하다는 것을 알 수 있다. 본 연구에서 제안할 프로그래밍 교수 전략이 개념 지도를 만들기 위한 좋은 방법이 될 수 있고, 실제 프로그래밍하는 과정이 개념을 형상화하는 과정이라고 볼 수 있고, 이 형상화하는 과정이 개념지도를 만드는 것으로 볼 수 있다. 프로그래밍에서는 실제 설계하는 구현하는 과정이라고 볼 수 있다.

2.3 프로그래밍 교육과 추상화

Nong Ye과 Gavriel Salvendy[10]은 컴퓨터 프로그래밍 지식을 5개의 추상화의 단계를 나누었다. 5개의 단계는 목적(objective), 개념(conceptual), 기능(functional), 논리(logical), 및 물리(physical)적인

추상화이다. 컴퓨터 프로그래밍 지식을 습득함에 따라 추상화의 단계가 변화한다는 것이다. 이 논문에서는 전문가와 초보자간의 추상화의 단계를 실험적으로 분석하였다. 실험결과는 전문가는 초보자보다 개념적인 지식과 기능적인 지식에서 추상적인 표현이 더 우수하고, 전문가는 구체적인 지식은 물리적인 단계에서 더 좋고, 논리적인 단계에서는 좋지 않다. 컴퓨터 프로그래밍 교육에서는 전문가가 추상적인 지식이 더 좋다는 것이다. 본 논문에서도 초등학교에서 컴퓨터 프로그래밍을 하는 것은 추상화의 능력을 기르는 역할을 해야 한다는 것이다.

3. 프로그래밍 교수 전략

3.1. 개요

본 논문에서는 초등학생을 위한 프로그래밍 교육을 하기 위한 교수 전략을 개발한다. 교수 전략은 주어진 문제를 기반으로 프로그래밍 언어 교육을 하면서 논리적인 사고 개발과 창의적인 사고 개발에 초점을 맞춘다. 논리적 사고 개발은 주어진 조건에서 간단히 표현하고 효율적으로 문제를 해결하는 방법을 만들어 보는 것이고, 창의적 사고 개발은 다르게 표현하는 방법과 빨리 문제 해결을 하는 방법에 초점을 맞춘다.

본 논문에서 제안한 프로그래밍 언어 교수 전략은 프로그래밍의 원리를 찾아서 그 전략을 마련한 것이다. 프로그래밍은 데이터와 알고리즘의 집합이다. 데이터 부분을 표현하는 입출력의 원리, 입반화의 원리가 필요하고, 알고리즘 부분을 표현하기 위하여 효율성의 원리, 단순화의 원리와 추상화의 원리가 필요하다. 이 원리들이 초등학생들의 프로그래밍 언어 학습을 위해서 필요하기 때문에 프로그래밍 언어 교수 전략을 제안한 것이다. 다음의 각 절에 이 원리들에 대해서 자세히 설명한다.

3.2. 전략1: 입출력 원리

초등학생을 위한 프로그래밍 언어 교육을 하기 위한 전략으로 입출력의 원리이다.

원리 1 (입출력 원리)
프로그래밍 원리는 입력한 정보를 처리하여 원하는 출력값을 만들어낸다.

교사가 학생들에게 필요한 문제를 출제할 때 반드시 입출력력의 원리에 적합한 문제를 내어야 한다. 반드시 입력 값을 학생들이 입력하거나 프로그래밍 내부에서 값을 생성할 수 있어야 한다. 출력 값을 화면에 표시할 수도 있고, 또는 데이터로 저장할 수도 있거나 다른 프로그래밍의 입력으로 될 수도 있다.

다음과 같은 문제를 살펴보자. 초등학교 5학년 수학 교과서에 나오는 최대공약수와 최소 공배수를 계산하는 프로그래밍을 생각한다. 만약 교사가 다음과 같은 문제를 내었다고 한다.

문제 1
최대 공약수와 최소 공배수의 값을 구하는 프로그래밍을 작성하시오.

이 문제는 입력 값이 무엇인지 명확하지 않다. 또한 출력 값을 어떻게 해야 하는지 명확하지 않기 때문에 입출력의 원리에 적합하지 않다. 따라서 이를 문제1-1처럼 변형하여 입출력의 원리에 적합하게 문제를 재구성할 수 있다.

문제 1-1
두수를 입력받아서 최대 공약수와 최소 공배수를 화면에 출력하는 프로그래밍을 작성하시오.

문제1-1은 입력 값이 두 숫자라는 것이 명확하고 출력하는 값인 최대 공약수와 최소 공배수를 화면에 출력하는 것이 명확하다. 이와 같이 교사가 문제를 출제할 때에 입출력의 개념을 학습하게 하기 위해서 문제에 입출력을 명확하게 기술해야 한다.

학생들은 주어진 문제에서 입력부분을 찾는 방법과 출력하는 부분을 찾는 방법에 대해서 계속 고민해야 하고, 또한 입력하는 방법과 출력하는 방법에 대해서 명확하게 하여야 한다.

3.3. 전략2: 효율성의 원리

초등학생을 위한 프로그래밍 언어 교육을 하기 위한 전략은 효율성의 원리이다.

원리 2 (효율성의 원리)
작성하는 프로그래밍의 처리를 빨리 수행하여 적은 자원을 사용하여 빠른 결과를 원한다.

컴퓨터는 입력된 데이터를 빨리 처리하여 원하는 결과를 빨리 도출하는 것이다. 이때 하드웨어 성능도 중요하지만 하드웨어 성능은 동일하다고 보고 알고리즘의 성능만 평가하는 것이다. 어떤 학생이 10명의 학생을 한 줄에 세워놓고 “정보”라는 학생을 찾을 때에 처음부터 한 사람이 비교하는 방법을 사용할 경우와 학생을 가나다라 순서대로 세워놓고 가운데부터 찾아가는 방법을 선택한 두 가지 경우가 있을 때에 뒤의 방법을 선택한 학생이 빨리 결과를 찾아낸다. 학생의 수가 100명, 1,000명이 될 때에는 더욱더 빠르다는 것을 느낄 것이다. 프로그래밍의 그 효율성이 매우 중요하고, 정보처리 속도가 매우 중요하다는 것을 알 수 있다.

교사가 문제를 출제할 때에는 효율성을 반드시 프로그래밍 문제에 강조해야 한다. 위의 문제 1-1에서 효율성을 강조하는 문제는 다음 문제 1-2와 같이 변경할 수 있다.

문제 1-2
두 수를 입력받아서 가능한 빠르게 최대 공약수와 최소 공배수를 화면에 출력하는 프로그래밍을 작성하시오.

문제1-2는 문제1-1에서 가능한 빠르게 등의 효율성의 문구가 내포되게 만들면 더욱 좋은 문제가 될 것이다.

3.4. 전략3: 일반화의 원리

초등학생을 위한 프로그래밍 언어 교육을 하기 위한 전략은 일반화 원리이다.

원리 3 (일반화의 원리)
입력 데이터 값의 범위를 확대하거나 입력 데이터의 수를 확대하는 것이다.

초등학생들은 학습을 통하여 일반화의 원리를 깨닫는 것은 매우 중요하다. 일반화를 하면서 학생들이 깊이 있는 사고를 하고 새로운 문제를 해결할 수 있는 능력이 생긴다. 두 개에서 최대값을 찾는 방법을 아는 학생이 세 개에도 최대값을 찾는 방법을 적용할 수 있고, 네 개, 다섯 개 등으로 확대할 수 있는 능력이 매우 중요하다.

이처럼 일반화하면서 많은 새로운 문제들을 창출할 수 있고, 새로운 해법을 알 수 있다.

문제 1-2를 좀 더 일반화한 문제를 표현하면 다음과 같은 문제 1-3이 된다.

문제 1-3
세 수를 입력받아서 가능한 빠르게 최대 공약수와 최소 공배수를 화면에 출력하는 프로그래밍을 작성하시오.

문제1-2는 두 개의 수를 입력받는 것에 세 개의 수를 입력받는 것으로 확장할 수 있고 좀 더 일반화하는 네 개의 수를 입력받는 것으로 될 수 있다. 이처럼 주어진 문제를 일반화할 수 있는 문제를 교사가 출제할 수 있어야 하고 또한 지도 할 수 있어야 한다.

3.5. 전략4: 단순화의 원리

초등학생을 위한 프로그래밍 언어 교육을 하기 위한 전략은 단순화 원리이다.

원리 4 (단순화의 원리)
어떤 문제를 해결할 때에 가능한 간단한 방법으로 해결하여 하는 것이다.

학생들이 프로그래밍을 할 때에 가능한 간단한 논리구조로 설명할 수 있어야 한다.

논리적인 구조를 간단히 표현하는 내용이 있는 문제를 만들기 위하여 문제1-2를 변경하여 문제1-4로 변경할 수 있다.

문제 1-4
두 수를 입력받아서 가능한 빠르고 간단한 논리적인 구조를 갖는 최대 공약수와 최소 공배수를 화면에 출력하는 프로그래밍을 작성하시오.

문제1-4에는 두 개의 수를 입력받아 가능한 빠르고 간단한 논리적인 구조로 표현할 수 있는 문제이다. 이 문제에서 논리적인 구조를 명확히 하기 위해서 반복문을 이용하여 등을 구체적으로 지시할 수도 있다. 이처럼 주어진 문제를 가능한 간단한 방법으로 프로그래밍을 할 수 있는 문제를 교사가 출제할 수 있어야 하고 또한 지도 할 수 있어야 한다.

3.6. 전략5: 추상화의 원리

초등학생을 위한 프로그래밍 언어 교육을 하기 위한 전략은 추상화 원리이다.

원리 5 (추상화의 원리)
 프로그래밍의 기본 구성 요소를 추상적인 용어로 표현하는 것이다.

프로그래밍을 할 때에 데이터 부분과 데이터를 처리하는 부분으로 나누어진다. 추상화의 요소에도 데이터를 부분을 추상화를 하는 방법과 데이터를 처리하는 부분을 추상적인 용어로 표현하는 능력이 필요하다. 데이터 부분은 데이터를 추상적인 용어로 잘 표현하여 조합하는 능력을 기르는 것이고, 데이터를 처리하는 부분은 간단한 명령어로 표현하는 부분과 같은 기능을 하는 논리적인 프로그래밍 단위들을 만들어서 이름을 명명하여 함수로 만들거나 블록으로 구조화하는 부분으로 나누어진다.

추상화의 능력을 향상하기 위한 문제를 만들기 위하여 문제1-4를 변경하여 문제1-5로 변경할 수 있다.

문제 1-5
 두 수를 입력받아서 가능한 빠르고 간단한 논리적인 구조를 갖는 최대 공약수와 최소 공배수를 화면에 출력하는 프로그래밍을 작성하시오. (단, 가능한 함수를 만들어서 사용하시오.)

문제1-5에는 함수를 지도하기 위하여 추상화의 원리를 학습하게 하는 것이다. 문제1-5는 명령어 그룹들을 함수로 표현하는 능력을 기르는 것이다. 따라서 문제1-5에서 추상화의 원리를 학습하기 위하여 함수를 이용하는 것을 명확하게 표시한다. 이처럼 주어진 문제를 해결할 때에 추상화의 원리를 학습하게 하기 위한 방법으로 프로그래밍 할 수 있는 문제를 교사가 출제할 수 있어야 하고 또한 지도 할 수 있어야 한다.

4. 타당성 검증

4.1. 전문가 조사 방법 채택 이유

본 논문에서 초등학생들의 논리력과 창의력 향상을 위한 검증 방법으로 전문가들의 심층 면접 방법을 채택 하였다. 이 방법은 델파이 방법의 일부 이용한 것

이다. 델파이 방법[9]은 전문가 집단에서 심층적으로 검증하는 방법으로서 미래 기술 및 경영 예측과 교육에 많이 사용하고 있는 방법이다. 이 방법은 기술적이고 현장의 데이터가 편중될 수 있는 사항에서 일반적인 설문조사보다 좋은 방법이지만 전문가들의 심층 면접을 전문 기술이나 방법에 대한 의견을 묻는 것은 타당하지 않다.

현재 컴퓨터 교육에 사용하는 교수 방법론에 대한 대부분의 교육학의 일반 교수 방법론을 프로그래밍 수업에 적용한 논문으로서 대부분 학교에서 학생들에게 수업한 결과를 가지고 단순한 통계 조사한 것들이 다[4]. 이 방법은 우리나라의 경우에 전국단위의 학생들을 수업에서 효과성 검증은 표본추출과 수업하는 교사의 특성에 따라 왜곡된 데이터가 많이 발생한다. 또한, 현실적으로 현장 검증 데이터는 몇몇 학교에 실험적으로 이용하는 것 또는 적용하는 사항에 데이터의 신뢰성에 문제가 생길 수 있기 때문에 본 논문에서 전문가들의 의견을 반영한 델파이 방법을 변형한 방법을 채택하였다. 델파이 기법은 교육 전문가들의 의견을 반영하여 타협하는 방법으로 반복적인 피드백을 통한 의견 수렴방법으로서 1948년 미국의 RAND연구소에서 개발되어 광범위하게 사용하는 연구 방법론이다[9]. 본 연구에서는 전문가들의 심층면접을 할 때에 델파이 방법의 절차만을 따르는 과정을 이용한 검증 방법을 채택하였다.

4.2. 검증 방법

검증 방법은 다음과 같은 5단계의 절차를 채택하였다.
 1단계에서는 전문가 집단을 선정한다.
 2단계에서는 설문조사 및 새로운 의견을 수렴한다.
 3단계에서는 설문 분석하여 평균 및 표준 편차 등을 계산한다.
 4단계에서는 평균과 표준 편차 등을 제한한 함과 동시에 2차 설문조사한다.
 5단계에서는 분석한다.

전문가 집단을 선정할 때에는 다음의 조건을 만족하는 전문가들을 선정하였다. 초등학교 교사는 초등학교 현장에서 실제 5년 이상 컴퓨터 교육을 실시한 경험이 있고 최소한 컴퓨터 교육을 석사학위를 받은 교사들로 초등학

교 교사 20명을 구성하였다. 측정 방법은 다음 <표1>과 같이 5단계 측도를 이용하였다.

<표 1> 설문조사 양식

전략	매우도움	도움된다	보통	도움안됨	매우도움안됨
입출력의 원리					
효율성의 원리					
일반화의 원리					
단순화의 원리					
추상화의 원리					

5단계 측정값은 제안한 프로그래밍 교수 전략이 각 항목별로 “매우 도움 된다”는 5점이고 “도움 된다”는 4점이고 “보통”은 3점이고, “도움 안된다”는 2점이고 “매우 도움 안됨”은 1점이다.

첫 번째 검증 과제는 입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리가 초등학생들의 논리적 사고 형성에 도움이 되는지를 검증하는 것이고, 두 번째 검증 과제는 이들 원리가 현재 초등학생들의 창의성 계발에 도움이 되는지를 검증하는 것이고, 세 번째 검증 과제는 이들 원리가 현재 초등학교의 ICT 교육 방법의 프로그래밍 언어 교육에 도움이 되는지를 검증하는 것이다.

4.3. 분석 결과

본 논문에서 제안한 초등학생들의 창의성 계발과 논리 계발을 위한 프로그래밍 언어 교수 전략 개발에 대한 타당성에 대한 분석 결과는 다음과 <표2>와 같다.

<표2> 1차 전문가 조사 결과

교수 전략	창의성 계발에 도움이 된다.		논리 계발에 도움이 된다.		초등학교에서 필요하다	
	평균	표준 편차	평균	표준 편차	평균	표준 편차
입출력 원리	4.05	0.69	4.55	0.60	4.65	0.67
효율성 원리	4.60	0.50	4.55	0.69	4.75	0.64
일반화 원리	4.40	0.68	4.55	0.60	4.75	0.55
단순화 원리	4.40	0.60	4.80	0.52	4.70	0.47
추상화 원리	4.30	0.80	4.40	0.82	4.25	0.79

1차 조사 결과를 분석하여 보면 다음과 같다. 첫 번째 검증과제인 “입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리가 초등학생들의 논리적 사고 형성에 도움이 된다.”에 대한 데이터 결과를 분석하여 보면 입출력의 원리의 중요성은 4.55점이고, 효율성의 원리의 중요성은 4.55점이고, 일반화의 원리의 중요성은 4.55 점이고, 단순화의 원리의 중요성은 4.80점이고, 추상화의 원리의 중요성은 4.40점이다. 이 자료에서는 초등학생들의 논리성 함양에는 단순화 제일 중요한 요소이다. 물론, 입출력, 효율성, 일반화 및 추상화의 원리도 중요한 요소이다. 초등학생들의 논리성 교육에 프로그래밍 교수 전략에 매우 중요한 요소라는 것을 알 수 있다.

두 번째 검증과제인 “입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리가 초등학생들의 창의성 계발에 도움이 된다.”에 대한 분석하여 보면 입출력의 원리의 중요성은 4.05점이고, 효율성의 원리의 중요성은 4.60점이고, 일반화의 원리의 중요성은 4.40점이고, 단순화의 원리의 중요성은 4.40 점이고, 추상화의 원리의 중요성은 4.30 점이다. 이 데이터를 자세히 보면 입출력의 원리는 창의성 계발에 다른 원리들보다 상대적으로 낮다. 단순한 입출력에 대한 프로그래밍 교육 전략은 창의성 계발에 다른 교육 전략보다 낮다는 의미이다. 그렇지만 평균적으로 의미 있는 교육 전략이라는 것을 알 수 있다.

세 번째로 본 논문에서 제안한 “입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리”가 교사들이 초등학생들의 프로그래밍 교육에 꼭 필요하기에 대한 질문 하였다. 이 결과에 대한 <표2>의 1차 분석은 다음과 같다. 입출력의 원리의 중요성은 4.65점이고, 효율성의 원리의 중요성은 4.75점이고, 일반화의 원리의 중요성은 4.75 점이고, 단순화의 원리의 중요성은 4.70점이고, 추상화의 원리의 중요성은 4.25점이다.

2차 전문가 조사 결과는 1차 전문가 결과의 각 항목별로 평균과 표준 편차 정보를 전문가들에게 보내고 자신의 설문지를 보내어서 다시 수정사항이 있으면 설문을 다시 접수 받았다. 2차 조사 결과는 다음 <표3>과 같다.

<표3> 2차 전문가 조사 결과

교수 전략	창의성 계발에 도움이 된다.		논리 계발에 도움이 된다.		초등학교에서 필요하다	
	평균	표준편차	평균	표준편차	평균	표준편차
입출력 원리	4.05	0.69	4.50	0.51	4.70	0.57
효율성 원리	4.85	0.37	4.65	0.67	4.75	0.64
일반화 원리	4.30	0.66	4.60	0.50	4.85	0.37
단순화 원리	4.40	0.60	4.90	0.31	4.85	0.37
추상화 원리	4.10	0.72	4.25	0.79	4.05	0.83

2차 전문가 조사 결과를 분석하여 보면 다음과 같다. 첫 번째 검증과제인 “입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리가 초등학생들의 논리적 사고 형성에 도움이 된다.”에 대한 데이터 결과를 분석하여 보면 입출력의 원리의 중요성은 4.50점으로 1차보다 조금 줄어들었다. 효율성의 원리의 중요성은 4.65점으로 1차보다 조금 증가하였다. 일반화의 원리의 중요성도 4.60점으로 1차보다 증가하였다. 단순화의 원리의 중요성은 4.90점은 1차보다 증가하였다. 추상화는 4.25점으로 1차보다 줄어들었다. 또한 표준편차가 1차보다는 2차가 줄어들었기 때문에 전문가들의 의견이 수렴하였다는 것을 알 수 있다. 이상의 자료를 정리하면 효율성, 일반화, 단순화는 상대적으로 입출력과 추상화보다 프로그래밍 교수 전략에 사용하면 학생들의 논리력 계발에 많은 도움이 된다는 것을 알 수 있다. 물론, 추상화와 입출력도 4점 이상이기 때문에 초등학생들의 논리력 향상에 의미 있는 프로그래밍 교수 전략이다.

두 번째 검증과제인 “입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리가 초등학생들의 창의성 계발에 도움이 된다.”를 분석하여 보면 입출력의 원리의 중요성은 4.05점으로 변동이 없고, 효율성의 원리의 중요성은 4.85점으로 많이 증가하였고, 일반화의 원리의 중요성은 4.30점으로 낮아졌고, 단순화의 원리의 중요성은 4.40점으로 변화 없고, 추상화의 원리의 중요성은 4.10점으로 낮아졌다. 이 부분에 초등학생들의 인지적인 단계에서 추상적인 개념을 배운다는 것이 매우 어렵기 때문에

창의성 계발에 다른 요소보다 중요하지 않음을 알 수 있다. 그렇지만 모든 요소들이 4점 이상이기 때문에 창의성 계발에 의미 있는 프로그래밍 교수 전략이 된다고 볼 수 있다.

세 번째로 본 논문에서 제안한 “입출력의 원리, 효율성의 원리, 일반화의 원리, 단순화의 원리 및 추상화의 원리”가 교사들이 초등학생들의 프로그래밍 교육에 꼭 필요한가에 대한 2차 전문가 분석 결과는 다음과 같다. 입출력의 원리의 중요성은 4.70점으로 높아졌고, 효율성의 원리의 중요성은 4.75점으로 변화 없고, 일반화의 원리의 중요성은 4.85점으로 높아졌고, 단순화의 원리의 중요성은 4.80점으로 높아졌고, 추상화의 원리의 중요성은 4.05점으로 낮아 졌다.

본 심층 분석으로 1차 분석보다 2차 분석의 표준편차가 적어진다는 것은 전문가들의 의견 수렴이 되었다는 것을 반증할 수 있다. 다만, 초등학교의 ICT 교육에서 각 교수 전략이 필요한가에 대한 추상화의 부분에서 표준편차가 조금 증가하였다. 이 부분에 대한 전문가들의 의견이 수렴되지 않는다는 것을 알 수 있다. 그러면 추상화도 논리성과 창의성 계발에 필요한 부분으로는 수렴되었다는 것을 알 수 있다.

5. 결론

본 연구는 초등학생들의 창의성과 논리 함양을 위한 프로그래밍 언어 교수 전략을 개발하였다. 본 연구에서 개발한 프로그래밍 언어 교수 전략은 입출력의 원리, 일반화의 원리, 단순화의 원리, 효율성의 원리, 추상화의 원리의 원리이다. 본 연구에서 개발 원리를 이용하여 초등학생들을 지도할 때에 문제를 설명할 때에 그 내용이 포함되어야 할 것이다.

본 연구에서 초등학교 교사들의 전문가 의견으로 수렴한 결과 개발한 프로그래밍 언어 교육 전략으로 창의성 계발을 위해 효율성의 원리, 단순화의 원리, 일반화의 원리, 추상화의 원리, 입출력의 원리 순으로 중요성이 나왔고, 논리성 함양을 위해 단순화의 원리, 효율성의 원리, 일반화의 원리, 입출력의 원리, 추상화의 원리 순으로 중요성이 나왔고, 초등학교 ICT 교육의 프로그래밍 언어 교육에 대해서

도 추상화의 원리를 제외하고 매우 필요하다는 것을 알 수 있었다.

본 논문에서 제안하는 프로그래밍 언어 교수 전략은 프로그래밍 교육 방법의 기본 철학을 제공하고자 한다. 또한 초등학교 현장 교사들이 본 논문에서 제안한 기본 철학으로 학생들에게 문제를 만들 수 있는 방향으로 설정하기 때문에 현장에 바로 적용할 수 있는 유용한 연구가 될 것이다.

참고문헌

[1] 김화경(2006), 재귀적 패턴과 거북 마이크로월드 설계, 한국수학교육학회지, 45-2, 165-176.

[2] 서울시교육청(2007), 정보통신기술소양교육 지도자료 서울: 형설출판사.

[3] 이미숙, 김갑수(2005). 초인지를 이용한 협력적 성찰 수업 모형의 개발 및 적용. 정보교육학회논문지. 9-2, 339-348.

[4] 이승현 김갑수(2008), 초등학생을 위한 프로젝트기반 프로그래밍 수업모형 개발 및 적용, 한국컴퓨터교육학회, 11-2, 23-34.

[5] Angela Carbone, Linda Mannila, and Sue Fitzgerald(2007), Computer science and IT teachers' conceptions of successful and unsuccessful teaching: A phenomenographic study, Computer Science Education, 17-4, 275-299.

[6] Dona M. Kagan(1989), Research on Computer Programming as a Cognitive Activity : implications for the study of classroom teaching, Journal of Education for Teaching, 15-3, 177-189.

[7] Hijazi, Sam; Smith, M. Leigh; Alvarado - Vazquez, Edgardo (2004), Creative Problem-Solving Software and Portals, Association of Small Computer Users in Education (ASCUE), 37th, Myrtle Beach, SC, June 6-10.

[8] Jeroen Keppens, David Hay(2008), Concept

map assessment for teaching computer programming, Computer Science Education, 18-1,31-42.

[9] Linstone,H.A and Turoff,M(1975). The Delphi Method: Techniques and Application, London: Addison-wesley.

[10] Nong Ye Gavriel Salvendy(1996), Expert-novice knowledge of computer programming at different levels of abstraction, Ergonomics, 39-3, 461-481.

[11] Salomon G & Perkins D.N(1987). Transfer of cognitive skills from programming : When and how?. Journal of Educational computing Research 3, 149-169.

[12] Scime(2008),Globalized computing education: Europe and the United States, Computer Science Education, 18-1, 43-64.

저자 소개

김 갑 수



1985.서울대학교 계산통계학과 졸업
 1987.서울대학교 전산과학 석사
 1996.서울대학교 전산과학 박사
 1987~ 1992 삼성전자 정보통신 연구소 과장
 1995~ 1998 서경대학교 전산정보학과 조교수

1998~현재 서울교육대학교 컴퓨터교육과 조교수~ 교수
 관심분야 : 이러닝, 정보영재교육, 컴퓨터교육, 소프트웨어 공학
 e-mail: kskim@snue.ac.kr