Journal of The Korea Society of Computer and Information
Vol. 22 No. 8, pp. 101-106, August 2017
www.ksci.re.kr
https://doi.org/10.9708/jksci.2017.22.08.101

Teaching-Learning Model for Programming Language Learning with Two-Step Feedback

Boseob Kwon*

Abstract

In this paper, we propose a new teaching-learning model with two-step feedback on programming language learning, which is a basic preliminary learning for programming. Programming learning is aimed at improving problem solving skills and thinking by experiencing problem solving through programming. For programming, the learner must know how to work with the computer and what to do with it. To do this, concrete thinking should be established and described in an accurate programming language. In recent, most studies have focused on the effects of programming learning and have not studied the effects of education on language itself. Therefore, in this study, the teaching-learning model for programming language education is presented and applied to the field, and the results are compared with the existing instructional- teaching model.

▸Keyword: Programming language Learning, Problem Solving, Feedback

I. Introduction

최근 세계적으로 ICT와 교육의 융합 관점에서 소프트웨어의 교육문제가 이슈로 부각되고 있으며 이는 우리들의 생존과 관련이 있는 문제로 볼 수 있다. 인공지능, 지능형 로봇, 사물인터넷, 빅데이터 분석 및 활동 등 4차 산업혁명시대로 대변된다. 이들은 소프트웨어를 통해 실현되기 때문에 소프트웨어 교육이 매우 중요하다[1].

해외 각국들은 소프트웨어의 중요성을 깨달아 경쟁적으로 코딩을 정규 교육과정에 반영해 교육중이다. 특히 영국 정부는 지난 2014년 가을학기부터 초·중·고교에서 소프트웨어를 필수 과목으로 지정하였으며, 우리나라도 전국 초·중·고교를 대상으로 2018년부터 소프트웨어 교육을 의무화 하고 있다[2][3].

소프트웨어 교육과 거의 같은 의미를 가지는 코딩 교육 혹은 프로그래밍 학습은 크게 두 가지로 구분된다. 프로그래밍 학습은 프로그래밍 언어 습득에 초점을 주는 관점과 고등인지 기능의 습득에 초점을 두는 경우로 나눌 수 있다. 전자는 프로그래밍 언어의 습득과 다양한 명령어들의 사용에 관계된 규칙의 이해 등을 강조하며, 학습의 목표로는 프로그래밍 언어의 명령어, 형식적인 절차 및 변수 등과 같은 개념의 이해와 사용으로 제한된다. 프로그래밍의 목적을 고등인지 기술향상에 중점을 두는 경우에는 프로그래밍 학습 과정에서 반성적 사고가 필요하다. 문제 해결과 계속적인 오류 검증 및 수정 등이 요구하는 반성적 사고를 통해 고등인지 기술향상에 중점을 둔다. 프로그래밍 과정에서 문제의 분석, 세분화, 해결 방안 모색 및 실행 등을 통하여 학습자의 사고력을 향상시킬 수 있는데, 이는 여러 가지 인지적 기능의 사용이 필요하기 때문이다[4]. 문제를 해결할 수 있는 능력 및 여러 상황에 대처할 수 있는 능력과 창의적이고 논리적인 능력을 프로그래밍 언어 교육을 통해 학생들에게 길러 줄 수 있다[5][6].

프로그래밍을 위하여 학습자가 컴퓨터에게 무엇을 하도록 지시할 것인가? 어떤 의도한 것을 실행시킬 것인가? 컴퓨터가 어떻게 작업을 시킬 것인가를 학습자가 이해해야 한다. 즉각적인 피드백이 있는 프로그래밍 학습은 자체로 구체적인 사고를 확립하고 정확한 표현을 필요로 한다. 프로그래밍 과정을 통하여 과제를 이해하고 분석하고 방법을 모색하면서 학습자가 프로그램을 실행하면 학습자가 보다 논리적이고 창의적인 사고를 할 수

• First Author: Boseob Kwon, Corresponding Author: Boseob Kwon
*Boseob Kwon(bxkwon@anu.ac.kr), Dept. of Computer Education, Andong National University
• Received: 2017. 07. 25, Revised: 2017. 08. 01, Accepted: 2017. 08. 16.
• This work was supported by a grant from 2016 Research Funds of Andong National University

있으며, 학습자 자신이 어떻게 사고했는지를 탐색할 수 있다.

프로그래밍의 학습은 문제해결 과정에서 문제를 분석, 조직, 구체화하는 방법 등을 학습자가 스스로 익히고 주어진 문제를 분석, 계획, 실험 및 수정하는 과정을 통해 논리적이고 체계적인 사고가 이루어지도록 학습 환경을 제공한다. 이러한 학습 환경에서 여러 가지 문제해결 방법을 적용하고 실험할 수 있는 환경을 학습자에게 제공한다[7][8].

학생들의 창의적이고 논리적인 능력 향상은 프로그래밍 학습 방법에 따라 다양하게 나타나지만 일반적으로 강의식 수업이 학교 현장에서 많이 사용되고 있다. 이는 창의력을 향상시키기에는 많은 제약 조건이 존재한다.

프로그램 언어 교육에 대한 국내의 연구는 프로그래밍 언어교육에 대한 교수-학습 이론들은 있지만 교수방법, 교수-학습모형, 학습 자료에 대한 연구가 부족하고 현장에 적용된 사례도 드물어 교육을 담당하는 교수 담당자나 교사들에게 어려움이 있다[9].

지금까지의 프로그래밍 언어에 대한 대부분의 연구들은 프로그래밍 언어교육을 통하여 얻을 수 있는 효과들에 관한 연구이며 프로그램 언어 자체에 관한 교육의 효과를 연구한 것은 아니다. 따라서 본 논문에서는 프로그래밍 언어 학습에 적합한 교수-학습 모형을 제시하고 학교 현장에 적용하여 기존의 지시적 교수-학습 모형과 비교하여 그 결과를 분석하고자 한다.

II. Programming Learning

1. Difficulties in programming learning

프로그래밍 교육의 학습 효과로는 학습자들의 자기 주도적으로 문제를 해결할 수 있는 능력, 논리적인 사고력, 여러 가지 상황에 대처할 수 있는 능력을 길러 줄 수 있다. 이러한 프로그래밍 학습은 크게 언어습득 자체에 중점을 두는 것과 고등인지 능력의 습득에 중점을 두는 두 가지로 나눌 수 있다. 특히 고등인지적 능력을 요구되는 프로그래밍 학습에서 많은 초보자들이 어려움을 겪게 된다. 프로그래밍 초보자가 학습하는 경우에 겪게 되는 어려움에 관해 살펴보면 다음과 같다.

프로그래밍 초보자들이 범하기 쉬운 오류들을 다음과 같다.

- 전략적 오류: 주어진 문제 해석을 잘못하였거나, 제대로 해석한 경우라도 문제 처리 전략을 잘못 사용하여 결과적으로 주어진 문제가 요구하는 결과를 도출하지 못하게 되는 오류이다.
- 함수적 오류: 특정한 일을 처리하는 프로그램을 구현하는데 있어서 발생하는 오류이다.
- 구현상의 오류: 타이핑 오류를 포함한 일반적인 프로그래밍에서 일어날 수 있는 오류를 말한다.

이러한 오류들은 소프트웨어에 대한 지식의 부족, 프로그래밍과 관련된 근본적인 문제 해결의 능력 부족 및 컴퓨터에 대한 자신감의 결여로 프로그래밍의 초보자가 갖는 어려움으로 분석되었다[5].

2. programming learning Objectives

여러 가지 문제를 해결하기 위해서는 교육목표에 대한 정의와 명확한 체계와 의미를 부여할 필요가 있다. Bloom은 학생의 발달 영역에 따라 인지적 영역 , 정의적 영역, 심동적 영역으로 분류하였고, 인지적 영역에서는 지식의 수준을 단순한 것으로부터 복잡한 난이도 순서로 지식, 이해, 적용, 분석, 종합, 평가인 6개 영역으로 구분하여 초·중등학교의 다양한 교과에 사용하고 있지만 프로그래밍 학습에 적용에 적절하지 않는 경우가 있다.

프로그래밍 학습에서 Bloom의 교육목표 분류법에 대한 문제점을 제시하였다[10][11][12].

- ‘이해’보다 ‘적용’이 더 높은 수준이지만 프로그래밍에는 지식을 말이나 글로 표현하는 것보다 익힌 지식을 적용하여 문제를 해결하는 것이 쉬운 경우가 있다.
- 프로그래밍 학습에서는 ‘종합’ 과 ‘평가’라는 영역의 분류는 거의 존재하지 않는다.
- 초보자인 경우 프로그램 개발 수준을 제시할 때 지식을 적용하는 수준에서 개발하는 것이 더 일반적이다. 따라서 초보자인 경우에는 ‘적용’이 최상위 난이도인 인지적 영역의 요소가 된다.

3. programming learning Objectives

3.1 Denotative instructional-teaching model

지시적 학습에서는 교사가 중심이 되어 프로그래밍 과제의 해결책을 찾으며 프로그래밍의 기본 명령어 및 개념들을 학습자에게 제시한다.

다양한 사고와 창의적이며 논리적 사고를 제한하는 지시적 수업 방법은 프로그래밍 과정에서 학생들의 개인적인 다양성을 나타낼 수 없으므로 학생들은 프로그래밍의 개념이나 기본적인 명령어들을 학습하기가 곤란한 경우도 있다. 또한 학습된 프로그래밍 문제도 학생들이 일반화시키고 다른 문제에 적용시키는 능력에 얼마나 향상되었는지도 의심스럽다. 특히 습득된 프로그래밍 학습이 문제전략에 관련된 프로그래밍이라면 더욱 일반화하기가 어렵다.

지시적 수업을 단계별 활동으로 나누어보면 다음과 같다.

- 문제파악 단계: 전시학습 확인, 동기 유발, 학습목표 제시 및 학습문제의 제시 등의 활동을 한다.
- 문제해결의 원리 제시단계: 원리 제시 단계에서는 프로그래밍 언어의 구조, 프로그래밍 방법을 구조화 및 문제 해결의 일반 원리를 제시한다.
- 문제해결의 시연 단계: 예제 중심으로 교사가 문제 해결을 시연하고 설명한다.
- 적용 및 일반화 단계: 교사에 의해 연습 문제 제시, 교사에 의한 문제 해결의 원리가 제공 및 학생에 의한 문제해결이 이루어지는 단계이다.

3.2 Guilded discovery instructional-teaching model

학생들에게 프로그래밍 학습이 자기 주도적으로 수행할 수 있는 환경을 만들어 주며, 교사에게는 학생들이 문제 해결 과정

에 의한 학습 진행 과정을 지속적으로 관찰하여 학생들로 하여금 문제 해결 능력을 향상시키는데 그 주안점을 두는 것이 안내된 발견식 수업이다. 이 수업 모형을 4단계로 나누어 살펴보면 다음과 같다.

● 문제의 이해 단계: 이 단계를 또 다시 두 가지로 나누어 볼 수 있다. 하나는 구체적 행동이며 다른 하나는 행동의 토론이다. 구체적 행동에서는 문제의식 및 문제 해결의 의지를 가지는 구체적인 행동이며, 행동의 토론에서는 교사가 질문을 하면 학생은 자신의 행동에 대해 구체적으로 이야기함으로써 교사의 질문에 답한다.

● 계획의 수립 단계: 행동을 표현하는 단계로 학습자들은 토론 내용을 활동지에 기록하고 주요한 핵심 요소를 산출하며 전체의 주제와의 관련성을 고려하는 활동이 이 단계에서 이루어진다.

● 계획의 실행 단계: 프로그래밍과 실행의 두 가지 활동을 하는데, 이 단계에서 표현의 결과를 프로그래밍 명령어로 전환하는 프로그래밍 활동과 작성된 명령어를 컴퓨터에서 실행하는 실행 활동으로 나누어진다.

● 반성 단계: 오류 확인, 수정, 확장 및 일반화를 하는 단계로써 오류확인 및 수정 과정은 학습 목표와 결과를 비교한다. 프로그램의 오류 확인과 수정하여 이를 토대로 학습자의 학습 결과를 관찰하고, 확장과 일반화 과정에서는 또 다른 해결 방안이 있는지 탐색하여 개념, 법칙 및 원리를 이해하여 다른 영역의 문제에도 응용해 보는 단계이다.

3.3 Exercise Instructional-teaching model

행동주의에 이론적 토대를 형성하고 있는 시범-실습 모형은 새로운 지식 및 기능 습득에 주로 사용하는 학습 이론이다. 학습자의 관찰 학습 방법과 행동주의의 특징인 자극과 반응의 연합에 근거한 반복학습과 피드백을 학습자에게 제공함으로써 학습의 효과를 극대화시킬 수 있는 방법이다. 반복적인 실행과 그 실행에 대한 피드백은 학습의 효과를 향상시키는 것을 이 학습 방법의 특징이다.

시범-실습은 학습자의 자기 주도적으로 학습에 참여도를 높여준다. 단점으로는 단계적 학습이 가능하지만 학습자마다 개인차가 존재하므로 이를 충분히 반영하지 못하고, 교사의 시범할 경우 학습자들의 주의집중 및 분위가의 통제가 어려움이 있을 수 있다[7].

III. The Proposed Teaching-learning Model

1. Design

본 연구의 적용 대상은 경상북도 중소도시에 소재하는 A대학 영재교육원의 정보과정의 초등학생이며, 영재교육원 입학시험 때 실시된 시험을 기준으로 동일한 수준의 두 집단으로 나누어 실시하였다. 대부분의 학생들은 C 언어에 대해 초보자이지만 많은 학생들은 다른 프로그래밍 언어를 사용해 본 적이 있는 학생들이다.

비교 집단과 실험 집단 두 집단으로 나누어 본 연구를 수행하였다. 각각의 교수-학습 모형별로 18시수씩 적용하였다. 한 집단에 사용된 교수-학습 모형은 실험이 끝날 때까지 그 모형을 지속적으로 사용했다.

수업이 끝난 직후 실시하는 형성평가는 지필고사로 실행하여 평가하였다. 문제 출제는 정보처리기능사 기출문제 중에 학습내용과 관련이 있는 것을 사용하였다. 교수-학습 모형에 따라 지식부분과 적용부분에 대한 학업성취도 변화를 알아보기 위하여 집단별로 학업 성취도 평가 결과의 평균치 차에 대한 유의도 검증을 하였다.

2. Hypothesis

프로그래밍 언어 교육에서 학습모형에 따른 학습 효과를 알아보기 위해 다음과 같은 가설을 설정하였다.

연구가설 1: 프로그래밍 언어를 학습할 경우 지시적 교수-학습 모형과 제안된 교수-학습 모형의 학업성취도에 유의한 차이가 있을 것이다.

연구가설 2: 프로그래밍 언어 학습모형에 따라 지식부분에서 학업성취도에 유의미한 차이가 있을 것이다.

연구가설 3: 프로그래밍 언어 학습모형에 따라 적용부분에서 학업 성취도에 유의미한 차이가 있을 것이다.

3. Teaching-Learning Model with Two-Step Feedback

3.1 Proposed instructional-teaching model

연구에 사용된 교수-학습모형은 그림 1과 같다.

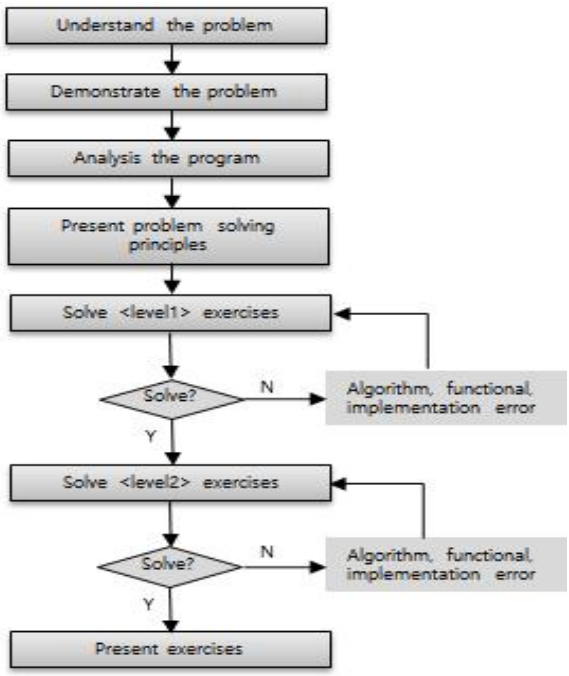

Fig. 1. Proposed teaching-learning Model

제안된 교수-학습모형의 절차는 다음과 같다.

● 문제 파악단계 : 전시학습의 확인, 동기유발 및 학습목표의 제시하는 단계이다.

● 문제 시연단계 : 교사의 시연에 따라 구체적 예제를 실습하고, 학습 목표의 구체화를 한다.

● 프로그램 분석 단계 : 학습자의 토의를 통하여 문제해결의 원리를 발견한다.

● 문제 해결 원리 제시단계 : 문제 해결의 일반 원리를 제공하고, 프로그래밍 언어의 구조 와 프로그래밍 해결 방안을 구체적으로 제시한다.

● <수준 1>,<수준 2> 연습문제 해결단계 : 문제 해결 원리 제시단계는 이미 학습한 문제 해결의 원리를 사용하여 문제를 해결한다. 프로그래밍 초보자들이 범하기 쉬운 오류는 교수자의 피드백을 통해 오류를 수정해 준다. 또한 연습 문제를 수준별로 제시하여 학습자가 다양한 형태의 과제들 접하여 볼 수 있게 하며 또한 학습자가 응용하여 문제를 해결할 수 있도록 한다.

제안된 교수-학습 모형은 교사 중심의 학습방법을 개선하여 프로그램 분석 단계에서 학습자가 좀 더 자기 주도적으로 학습에 참여할 수 있도록 한 지시적 교수-학습 모형이며, 연습 문제와 응용문제를 이용하여 다양한 사고를 할 수 있도록 하였다. 피드백에 의한 정보 제공을 함으로써 학습자의 행동을 수정하고 스스로 생각해 볼 수 있도록 한 교수-학습 모형이다.

3.2 Learning Procedure

학습 프로그램은 초보자가 C 프로그래밍 언어를 습득하는데 그 목적이 있으며, 지시적 교수-학습 모형과 제안된 교수-학습 수업 모형으로 제시한 두 가지로 제작되었다. 제안된 교수-학습 모형의 학습 과정을 살펴보면 다음과 같다.

● 학습목표를 확인하고 전시 학습을 상기한다.

● 교사가 학습자에게 여러 문제를 시연하여 한다.

● 프로그램 분석 과정은 학생들의 토의를 통하여 이루어진다.

● 문제를 시연한 후 학습자가 이를 통하여 그 문제에 대한 토의를 거친 후 교사가 문제 해결의 핵심 원리를 명확하게 제시한다.

● 앞 단계에서 습득된 문제해결의 원리를 사용하여 실제 연습문제 및 응용문제를 풀어본다.

IV. Result and Analysis

프로그래밍 언어 학습인 경우 교수-학습 모형에 따라 지식부분과 적용부분에 대한 학업성취도의 변화가 어떻게 다른지를 검증하기 위해 실험집단별로 차이가 있는지를 알아보는 연구이다. 연구방법에 따른 실험별로 수행한 결과를 독립표본 t-검증을 사용하여 분석하였으며 결과를 살펴보면 다음과 같다.

1. Difference of academic achievement according to teaching-learning model

교수-학습 모형에 따른 학업 성취도 차이는 독립표본 검정 결과 학업 성취도에 있어서 통계적으로 유의미한 차이를 보이고 있다(p <0.1).

Table 1. Difference of Academic Achievement according to Teaching-Learning Model

	N	average	Standard Deviation	t-value	p-value
Directive Model	15	42.3333	12.2280	1.846	0.076
Proposed Model	16	52.1875	17.2210		

학습자의 학업 성취도의 차이를 비교하면, 지시적 교수-학습 모형에서는 평균이 42.3333이고, 제안된 모형에서의 평균이 52.1875으로 나타났다. p값은 0.076이다. 이는 지시적 교수-학습 모형으로 학습한 경우와 제안된 교수-학습 모형으로 학습한 경우 학습자의 학업성취도는 유의확률 10%에서 유의미한 차이를 보이고 있다.(p<.1) 두 교수-학습 모형의 학업 성취도의 평균에 관한 표준편차는 지시적 수업 모형은 12.2280, 제안된 수업모형은 17.2210이다. 이는 프로그래밍 언어 학습에 대한 개인별 수준차가 크게 나타나 표준편차가 다르게 나타난 것으로 보인다. 또한 일부 학습자들이 선행학습을 통해 프로그래밍 학습이 어느 정도 이루어진 것으로도 보인다. 두 집단의 평균의 차는 약 10점, 표준편차의 차는 약 5점으로 제안된 교수-학습 모형의 평균이 지시적 교수-학습모형보다 높고 표준편차가 더 크게 나타났다.

2. Differences in academic achievement in knowledge

프로그래밍 언어 교수-학습 모형에 따라 지식부분에서의 학업성취도 차이를 t-검증으로 분석한 결과 p값이 0.207로 지시적 교수-학습 모형으로 학습한 경우와 제안된 교수-학습 모형으로 학습한 경우 지식부분에서 학업성취도에는 통계적으로 유의한 차이가 없는 것으로 나타났다.(p>.1)

Table 2. Difference of Academic Achievement in Knowledge according to Teaching-Learning Model

	N	average	Standard Deviation	t-value	p-value
Directive Model	15	13.6667	7.4322	1.293	0.207
Proposed Model	16	16.8750	6.2915		

이는 프로그래밍 언어 학습에 대한 선행 학습 등으로 인해 개인차가 크게 나타나 표준편차가 평균의 차이보다 크게 차이가 나타나 평균의 차이가 집단별 간의 차이에 크게 의미가 없다고 분석되

었기 때문이다. 또한 교수-학습의 대상이 일정 수준 이상의 학생으로 이루어져서 지식부분의 학습이 교수-학습의 모형의 차이에 대한 영향을 크게 받지 않은 결과일 수도 있다.

3. Differences in academic achievement in application

프로그래밍 언어 교수-학습 모형에 따라 적용부분에서의 학업성취도 차이를 분석한 결과 p값이 0.088로 유의수준 10%에서 학습자의 학업성취도에는 통계적으로 유의한 차이가 있었다.(p<.1)

Table 3. Difference of Academic Achievement in Application according to Teaching-Learning Model

	N	average	Standard Deviation	t-value	p-value
Directive Model	15	28.6667	7.6687	1.775	0.088
Proposed Model	16	35.3125	12.7107		

지시적 학습모형의 적용부분의 평균은 28.6667점이고 제안된 학습모형의 적용부분의 평균은 35.3125로 나타나 두 집단의 평균의 차는 약 7점으로 나타났다. 표준편차 또한 지시적 학습모형은 7.6687, 제안된 학습 모형은 12.7107로 나타나 두 집단 모두 표준편차가 큰 것으로 나타났으며, 이것은 프로그래밍 언어 학습에 대한 학습자의 개인차에서 비롯된 것으로 보인다.

V. Conclusions

프로그래밍 언어 학습에서 교수-학습 모형에 따라 현재 학교 현장에서 많이 사용되고 있는 지시적 수업모형과 논문에서 제안한 교수-학습 모형을 모형에 따른 학업 성취도의 차이, 지식, 적용부분에서의 성취도의 차이를 비교하였다.

프로그래밍 언어 학습에서 교수-학습 모형에 따른 학습자의 학업성취도의 차이는 두 집단 간의 유의미한 차이가 있음을 유의 수준 10%에서 검증하였다. 학업성취도의 차이에 대한 분석 결과에서 표준 편차가 매우 크게 나타났다. 이는 학습자들이 비교적 일반 학생들과 비교하여 영재교육원에 입학할 정도의 우수한 학생일 뿐만 아니라 프로그램에 대한 학습자의 관심과 선행학습으로 인해 프로그래밍에 대해서는 어느 정도 우수한 집단으로 볼 수 있다. 이로 인해 프로그래밍 언어 학습에 대한 학습자간 개인차가 존재하여 표준 편차의 점수가 크게 나타났다.

프로그래밍 언어 교수-학습 모형에 따른 지식부분에서 학습자의 학업성취도 차이는 두 집단 간의 유의미한 차이가 없는 것으로 나타났다. 적용부분에서는 학습자의 학업성취도는 유의미한 차이가 있었다.

이러한 연구 결과를 종합하면 프로그래밍 언어 학습에서는 학습자간 개인차가 크게 나타나는 것으로 볼 수 있으나 본 연구의 대상 집단이 연구에 사용된 C언어에 대해서는 초보인 경우학습자들이 대다수이지만 처음 C 언어를 학습하는 학습자라 할지라도 다른 프로그래밍 언어에 접해본 적이 있는 학습자가 대부분이므로 표준 편차에 영향을 미친 것으로 보인다. 본 연구가 프로그래밍 언어 자체에 대한 학습이므로 문법을 중심으로 학습하였다. 이로 인해 초보자인 경우 프로그래밍 언어의 지식을 말이나 글로 표현하는 것보다 익힌 지식을 적용하여 문제를 해결하는 것이 쉬운 경우가 있을 수 있다. 또한 학습자간 개인별 차이가 있으므로 프로그래밍 언어 교육에서는 집단을 구성하여 학습하는 경우 개별학습이 가장 효과적일 것으로 생각할 수 있다.

REFERENCES

[1] etnews, http://www.etnews.com/20170421000360
[2] Jung-Sook Sung and Hyeon-Cheol Kim, “Analysis on the International Comparision of Computer Education in Schools”, The Journal of Korean Association of Computer Education, Vo. 18, No. 1, pp. 45-54, Jan, 2015.
[3] Hyun-Jong Choe, Tae-Ok Song and Tae-Wuk Lee, “Comparative Study of Imformatics Subject Curriculums and Textbooks in Middle School Between Korea and England”, The Journal of Korean Association of Computer Education, Vol. 21, No. 2, pp. 145-152. Feb. 2016.
[4] Sangjin An and Youngiun Lee, “A Stusy of Programming Interest”, Proceeding of the Korean Association of Computer Education, Vol. 17, No. 1, pp. 103-106, 2013.
[5] SooHwan Kim, SeonKwan Han and HyeonCheol Kim, “Analysis of Programming Processes Through Novices' Thinking Aloud in Computational Literacy Education“, The Journal of Korean Association of Computer Education, Vol. 14, No. 1, pp. 13-21, Jan. 2014.
[6] Boseob Kwon, “The Effect of Cooperative Learning on Problem Solving in Programming Learning”, Journal of the Korea Contents Association, VOl. 14, No. 6, pp. 491-498, June, 2014.
[7] Yoon Young Kim and Yung Sik Kim, “Computer Education Curriculum and lnstruction : Effect of Learning a Divide-and-conquer Algorithm on Creative Problem Solving”, Proceeding of the Korean Association of Computer Education, Vol. 16, No. 2, pp. 9-18, 2013.
[8] Eunsoon Kim, Byungho and Heeok Heo, “The Effects of peer Tutoring on Academic Achievement and Peer Relationship in Programming Language Learning”, The Journal of Korean Association of Computer Education, Vol. 7, No. 4, pp. 111-119, July 2004.
[9] ByeongGeon Yu, JaMee Kim and WonGyu Lee, “Analysis

on the Relation Between Programming Achievement and Problem Solving According to Gender", The Journal of Korean Association of Computer Education, Vol. 15, No. 6, pp. 1-10, Nov. 2012.

[10] Hyun-Jong Choe, "Analysis about Learning Objectives of Informatics Textbooks in High School using Anderson's and Fuller's Taxonomy of Educational Objectives", Journal of the Korea Society od Computer and Infomation, Vol. 19, No. 9, pp. 185-196, Sept. 2014.

[11] Hyun-Jong Choe, "Study of Analysis about Learning Objectives of Informatics Textbooks in Middle School using Anderson's and Fuller's Taxonomy of Educational Objectives", The Journal of Korean Association of Computer Education, Vol. 17, No. 1, pp. 51-63, Jan. 2014.

[12] Fuller. et al., "Developing a Computer Science- Specific Learning Taxonomy", ACM SIGCSE Bulletin, Vol. 39, No. 4, pp. 152-170, 2007.I. Introduction

Authors

Boseob Kwon received the B.S. degree in Electric Engineering from Kyungpook National University, Korea, in 1983. and the Ph.D degree in Computer Science from Korea Advanced Institute of Science and Technology(KAIST), Korea, in 1997. He worked as a Senior Researcher with the Electronics and Telecommunications Research Institute from 1985 to 1998, After he is currently a Professor in the Department of Computer Education, Andong National University. He is interested in Computer education and Information Security.