

문제해결학습을 적용한 소프트웨어 교육 효과 분석

이영석

강남대학교 KNU 참인재대학 교양교수부

Analyzing the effect of software education applying problem-solving learning

Youngseok Lee

KNU College of Liberal Arts and Sciences, Kangnam University

요 약 다양한 문제 상황을 정확히 파악하고 컴퓨팅 기술을 이용하여 문제를 보다 효과적으로 해결하는 능력은 앞으로 더 중요해질 것이다. 이러한 배경으로 컴퓨팅 사고력과 함께 문제해결 능력을 갖춘 미래 인재 양성이 대학 교육에서 강조되고 있다. 따라서 본 논문에서는 컴퓨터 비전공자를 대상으로 컴퓨팅 사고력을 향상시키기 위한 소프트웨어 교육을 문제 해결 학습 기반으로 실시하여 효과를 분석하였다. 그 결과, 학생의 흥미와 배운 정도, 학업성취도에서 유의미한 차이가 나타났음을 알 수 있다. 따라서 학습 동기와 학습 방법에 대한 이해를 바탕으로 학생 수준에 적합한 실생활에 관련된 문제 해결 학습을 통해 소프트웨어 교육을 진행한다면 학생들의 관심을 유도할 수 있고, 학습 효과가 높아져서 컴퓨팅 사고력 향상에 도움이 될 것으로 기대된다.

주제어 : 문제 해결 학습, 소프트웨어 교육, 학습 효과, 학습 만족도, 컴퓨팅 사고력

Abstract The future society will be identify various problem situations accurately and the ability to solve problems effectively using computing technology become more important. Based on this background, an education of future human resources based on computational thinking as well as a problem-solving ability are important in university education. Therefore, in this paper, we have analyzed the effects of software education to improve computational thinking based on problem-solving learning. We have found that there is a significant difference between the interest of the students and their learning and academic achievements. Therefore, based on the understanding of the learning motivation and method, if the problem-solving learning is conducted in real-life scenarios suitable for the level of the student, it can be possible to induce the interest of the students and improve their computational thinking ability.

Key Words : Problem-solving learning, Software education, Learning effect, Learning satisfaction, Computational thinking

1. 서론

현대 사회는 소프트웨어 중심의 시대로 학문 간의 융합과 더불어 많은 분야에서 변화가 일어나고 있다. 이러한 변화에 맞춰 우리나라에서도 2015 개정 교육과정을 바탕으로 소프트웨어 교육을 강화하기 위하여 중학교 정

규 필수 교육과정에 편입시키고 대학에서도 소프트웨어 교육을 강화하고 있는 추세이다[1].

우리나라뿐만 아니라 미국, 영국, 일본 등의 주요 선진국들도 4차 산업혁명의 거센 변혁의 물결을 위기가 아닌 기회로 삼기 위해 컴퓨터 비전공자를 포함한 모든 학생들에게 소프트웨어 교육을 경쟁적으로 실시하고 있다[2].

*This Research was Supported by Kangnam University Research Grants.(2016)

*Corresponding Author : Youngseok Lee (yslee38@kangnam.ac.kr)

Received February 26, 2018

Accepted March 20, 2018

Revised February 28, 2018

Published March 28, 2018

융합적인 사고를 바탕으로 창의적인 인재를 양성하기 위해서는 이러한 소프트웨어 교육이 필수적이며, 변화하는 시대에 대비하여 컴퓨팅 사고력을 가진 인재를 양성하기 위한 다양한 교수 학습 방법에 대한 연구가 필요한 시점이다[3, 4].

하지만, 현실적으로 컴퓨터 비전공자들은 컴퓨터 활용 능력이나 오피스 활용 능력이 더 필요하다고 느끼며, 교육용 프로그래밍 언어 기반의 소프트웨어 교육에 대해서는 어려움과 함께 필요성을 공감하지 못하고 있어, 학생들에게 흥미를 유도하면서 컴퓨팅 사고력과 문제해결력을 향상시키는 것은 쉽지 않다[5].

문제해결력을 함양하기 위해서는 학생들에게 실생활에 기반을 둔 문제를 제시하고, 학생으로 하여금 제시된 문제를 확인하고 탐색하는 과정을 거쳐, 교육용 프로그래밍 언어를 도구로 문제 해결 방안을 구현하는 일련의 과정에서 학습이 이루어지도록 구성하여야 한다[6].

따라서 본 논문에서는 대학에서 소프트웨어 교육을 원활하게 수행하기 위하여 학생들에게 흥미를 유발하면서 컴퓨팅 사고력을 향상시킬 수 있는 문제해결학습 기반의 소프트웨어 교육 사례를 제시하고 그 효과를 분석하고자 한다.

2. 이론적 배경

2.1 소프트웨어 교육

소프트웨어 교육은 컴퓨터와 대화할 수 있는 프로그래밍 언어를 배워서 본인이 생각하는 절차와 방법대로 컴퓨터에게 명령을 내리고 그 결과를 확인하는 것이다. 이때, 소프트웨어 개발자를 양성하는 전문 프로그래밍 교육과, 컴퓨팅 사고력 향상과 문제해결력을 갖추기 위한 보편적인 프로그래밍 교육 등으로 구분할 수 있다. 본 논문에서는 4차 산업 혁명 시대의 미래 사회를 대비해서 컴퓨팅 사고력과 문제해결력을 갖춘 인재를 양성하기 위한 보편적 프로그래밍 교육에 초점을 맞춘다[7].

프로그래밍은 주어진 문제를 해결해 나가는 과정에서 문제를 분해하고, 그것을 해결해 나가기 위한 방법을 찾고, 점진적으로 이 과정을 반복 및 조합해 나가면서 작업을 완성해 나간다[8]. 이러한 관점에서 프로그래밍을 통해 정보 처리 능력, 절차적 사고 능력, 문제 해결력, 논리적 사고력, 추론 능력 등 다양한 능력이 향상되는 것이며, 이 과정에서 컴퓨터 비전공자들은 전혀 다른 새로운 시

각에서의 접근을 배우기 때문에 다양한 어려움에 직면하게 된다. 이 어려움을 줄이기 위해서는 본인이 해결할 수 있는 쉬운 문제 상황을 시작으로 점차 다양하고 복잡한 어려운 상황을 풀 수 있도록 구성하는 것이 필요하다.

2.2 문제 해결 학습

문제 해결 학습은 문제 중심 학습 또는 문제 기반 학습과 유사한 의미로 사용된다[9]. 학습자에게 문제를 제시하고, 제시된 문제의 해결방안을 탐색한 뒤, 개별학습 혹은 협력학습을 통해 해결방안을 마련하는 일련의 과정에서 학습이 이루어지는 학습 방법이다[10].

문제 해결 학습은 문제의 이해, 문제 해결 계획, 실천, 실천방법 수정 및 재수행을 실시하면서 문제를 해결해 낼 수 있는 능력을 향상시키는 것이라고 할 수 있다[6, 11]. 다양한 선행 연구에서 실시된 문제 해결 학습 기반 교육용 프로그래밍 언어 교육 단계를 바탕으로 하여 본 논문에서 제안하고자 하는 소프트웨어 교육에 필요한 문제의 특징을 다음과 같이 제안한다[12, 13].

- 실제 상황을 묘사하거나 해결하려는 문제
- 해결 방법이 한 가지 이상인 문제
- 지식의 전이가 가능한 문제
- 동기를 부여하고 도전의식을 불러일으키는 문제
- 확장 및 분해가 가능한 문제

일반적으로 프로그래밍 과정은 문제 해결을 하는 과정과 비슷하므로 본 연구에서는 다양한 분야에서의 문제 해결학습 관련 연구를 분석하여 소프트웨어 교육에 적용하고자 한다[14, 15].

3. 문제 해결 학습 적용

3.1 문제 해결 학습의 절차

문제 해결 학습을 실시하기 위하여 학습 절차를 정리한 결과는 Fig. 1과 같다[8]. 기본적인 프로그래밍 내용을 소개한 뒤, 학생에게 문제의 배경과 함께 문제를 제시한다. 학생들은 해결해야 하는 문제가 무엇인지, 이를 해결하기 위해 어떤 내용들이 필요한지 파악하고, 문제 해결이 가능한지 판단한다. 문제를 이해하지 못하면, 다시 기본 요소 이해 단계로 돌아가고, 문제에 대한 확인과 판단이 가능하면 풀기 위한 학습 내용이나 추가 사항을 도출하여 학습한다[8].

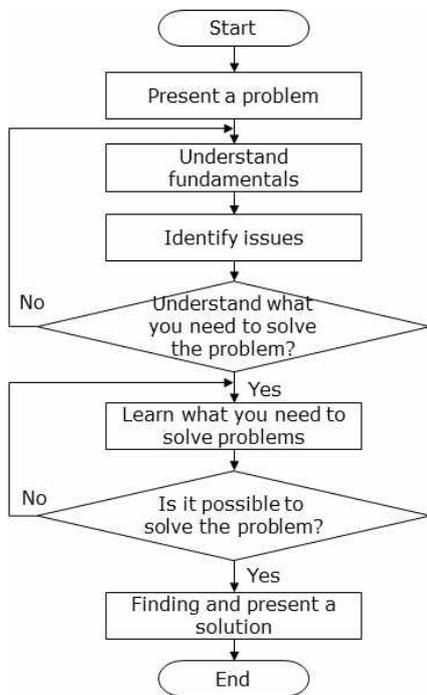


Fig. 1. Procedures for problem-solving learning

문제 해결이 가능하면 문제 해결안을 정리하고 발표 하지만, 문제를 풀지 못하면 다시 문제 풀기 위한 사항들을 추출하고 학습하도록 한다[8, 11].

각 문제는 수업 목표, 문제 내용, 수업 시간 등을 고려하여 배치하였으며, 강의 초반에는 문제를 아주 단순하게 제시하여 쉽게 해결하고, 점차 복잡한 문제를 풀 수 있도록 하여 학습 목표를 도달할 수 있도록 지속적으로 유도하고 반복 학습을 실시하였다[14]. 기본 강의계획안은 [12]에서 제안한 컴퓨터 비전공자들을 대상으로 하는 프로그래밍 교육과 유사하지만, 문제 해결 기법을 적용할 수 있도록 교육내용을 변형하였다.

3.2 사례 1 : 성적처리 프로그램

3.2.1 수업 목표

성적 처리 프로그램의 수업 목표는 3가지로 설정하였다. 첫째, 입·출력문을 여러 가지 형식에 맞게 사용할 수 있다. 둘째, 인간이 이해하기 쉬운 알고리즘을 컴퓨터가 해결할 수 있는 알고리즘으로 변환할 수 있다. 셋째, Python 혹은 스크래치를 사용할 수 있다.

3.2.2 문제 내용 및 진행 방안

문제 해결 학습에서 사용할 문제 내용은 다음과 같다.

학기가 종료되었을 때 필요한 성적 처리 프로그램을 만들고자 한다. 성적 처리 프로그램의 필요 요소로는 교과목명을 입력한 후, 출석, 과제, 시험 점수를 입력하면, 총점과 평균, 최고 점수, 최저 점수 및 학점을 알려주는 프로그램을 작성하고자 한다.

이 문제는 4주~11주 강의시간 동안에 지속적으로 실시하며, 중간고사 이전에 입출력 관련 내용 2주, 조건문 2주를 학습하고 부분적인 문제를 해결하고, 중간고사 이후에 반복문 2주, 자신만의 프로젝트를 위한 추가 1주를 제공하여 문제해결을 할 수 있도록 하였다.

3.3 사례 2 : 반복 합계 구하기

3.3.1 수업 목표

반복 합계 구하기 프로그램은 반복문을 학습하는 동안에 별도로 진행될 수 있도록 구성하였다.

입·출력문과 반복문을 여러 가지 형식에 맞게 사용할 수 있다.

3.3.2 문제 내용 및 진행 방안

문제 해결 학습에서 사용할 문제 내용은 다음과 같다.

사용자가 합계를 구하기 위해 시작값과 종료값을 입력하면 시작값부터 종료값까지 합계를 구한다. 이 방법을 활용하여 거북이 모듈을 활용하여 자신만의 그림 그리기 문제를 해결하도록 한다.

앞의 성적 처리 프로그램과 함께 6주~11주의 강의 시간에 반복 합계 구하기와 다양한 도형 그리기를 병행해서 진행하였고, 전체 5주 동안에 다양하게 시도해 볼 수 있도록 하였으며, 입·출력문 2주, 반복문 2주, 추가 문제 해결을 위한 1주를 제공하였다.

3.4 사례 3 : 계산기 프로그램

3.4.1 수업 목표

계산기 프로그램은 프로젝트 형태로 많은 프로그래밍 요소들이 결합되어 있으며, 수업 목표는 다음과 같다.

연산자 입력 후 계산기가 동작되도록 입·출력문, 조건문, 반복문을 여러 가지 형식에 맞게 사용할 수 있다.

3.4.2 문제 내용 및 방안

문제 해결 학습에서 사용할 문제 내용은 다음과 같다.

사용자가 2개의 숫자를 입력하고, 연산자를 입력하면 계산기와 동일하게 계산이 되어서 출력되도록 한다. 기본적으로 덧셈, 뺄셈, 곱셈, 나눗셈을 계산하도록 하고, 사칙연산이 익숙해지면, 거듭제곱 혹은 나머지 구하는 모듈러 연산 등을 통해 소수 구하기와 약수 구하기 등을 추가한다.

학기 중반 이후인 12~14주 사이에 진행하였고, 입출력 1주, 반복문 1주, 조건문 1주, 추가 문제 해결을 위한 1주를 학습하면서 자신만의 계산기 프로그램을 만들 수 있도록 하였다.

4. 소프트웨어 교육 효과 분석

4.1 연구 대상

2017학년도 K대학교 1학년을 대상으로 Python 언어 기반의 소프트웨어 교육을 실시하였으며 대상 학생에 대한 정보는 Table 1과 같다.

Table 1. Status of participants

Department	Control group	Experimental group
Social Welfare	32	34
Global Talents	32	54
Business Administration	47	72
Education	26	28
Total	137	188

부분 응답을 하지 않는 등 비정상적인 형태의 응답을 한 학생을 제외하고, 사회복지학부 학생 66명, 글로벌 학부 학생 86명, 경영관리 대학의 경제세무학과 47명과 경영학부 72명, 사범대학의 교육학과 14명, 유아교육과 26명, 특수교육과 14명을 대상 학생으로 하여 각 집단별 동질성 검증을 실시하여 집단간 차이가 없음을 확인하였다.

4.2 검사 도구

소프트웨어 교육의 효과를 분석하기 위해서 [6]의 연구에서 사용한 검사지를 번안하여 사용하였으며, 설문 문항에 대한 신뢰도는 Table 2와 같다. 모든 설문 문항에서 0.7 이상으로 높은 신뢰도가 나타났다.

Table 2. Reliability statistics

Survey question	Reliability coefficient
No.1 : Acquire new knowledge	.810
No.2 : What you learned	.802
No.3 : Structure of learning contents	.805
No.4 : Homework satisfaction	.802
No.5 : Test satisfaction	.784
No.6 : Teaching method	.782
No.7 : Interaction with professors	.783
No.8 : Credit Rating Method	.785
No.9 : Classroom environment	.802
No.10 : Course progress speed	.791
No.11 : Lecture grade	.790
No.12 : Attendance	.804
No.13 : Hope to have additional software education	.817
Total	.801

문항의 타당도 검사를 위해서 요인 분석을 실시한 결과는 Table 3과 같다. KMO and Bartlett's Test 수치가 .872로 높게 나타나면서 유의확률이 .000이어서 문항의 타당성은 확보되었다. 설문 문항이 유사하거나 동일한 내용을 묻는 사항은 아니므로 요인 분석 결과는 추가로 제시하지 않는다.

Table 3. KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.872
Bartlett's Test of Sphericity	Approx. Chi-Square	1586.782
	df	190
	Sig.	.000

4.3 분석 결과

2017학년도에 실시한 강의를 통해 나타난 학업 성과 출석 체크, 그리고 설문조사를 실시하여 사전·사후 검사를 실시하였다. 분석한 주요요소는 학생이 느낀 지식 습득 정도(No1), 얼마나 배웠는지 여부(No2), 학습 내용 구성(No3), 과제 만족도(No4), 추가 소프트웨어 교육 희망 여부(No13)이고, 이를 검증하기 위해서 실제 학생들이 수행한 학업 성적(No11)과 출결(No12)을 분석하였다.

출결 여부에 대한 사전·사후 독립표본 검정을 실시한 결과는 Table 4와 같다.

Table 4. Attendance pre-post test result (p<.05)

Elements		N	Avg.	Std.	t	P
No.12 : Attendance	Pre	137	4.57	.812	-.256	.798
	Post	188	4.59	.668		

출결 여부는 문제 해결 학습을 수행한 여부와 상관없이 유의미한 차이가 나타나지 않는다. 하지만, 그 외의 다른 부분에서는 유의미한 차이가 나타났다. 학생들의 수업 만족도 조사 결과는 Table 5와 같다.

Table 5. Satisfaction survey pre-post test result ($p < .05$)

Elements		N	Avg.	Std.	t	P
No.1 : Acquire new knowledge	Pre	137	3.35	1.122	-5.004	.000
	Post	188	3.93	.945		
No.2 : What you learned	Pre	137	3.53	.978	-3.734	.000
	Post	188	3.93	.904		
No.3 : Structure of learning contents	Pre	137	3.62	.964	-5.229	.000
	Post	188	4.13	.800		
No.4 : Homework satisfaction	Pre	137	3.07	1.034	-3.901	.000
	Post	188	3.51	.973		

지식 습득 정도의 평균은 통계적으로 $p < 0.05$ 수준에서 유의한 차이($t = -5.004$, $p = 0.000$)가 있는 것으로 나타났다. 따라서 학생들에게 문제 해결 학습을 실시하는 것이 학생이 느끼기에도 지식 습득에 효과가 있는 것으로 나타났다.

마찬가지로 얼마나 배웠는지 여부($t = -3.734$, $p = 0.000$)와 학습 내용 구성($t = -5.229$, $p = 0.000$), 과제 만족도($t = -3.901$, $p = 0.000$)도 모두 유의미한 차이가 있는 것으로 나타났다. 즉, 문제 해결 학습을 실시한 집단에서 지식 습득 정도와 학습 만족도, 학습 구성과 과제 모든 면에서 만족스러운 결과가 나타났다.

만족도 조사 결과와 별도로 학기 종료 후에 나타난 학생들의 성적과 함께 추가 소프트웨어 학습 희망 조사 결과는 Table 6과 같다.

Table 6. Academic achievement pre-post test result ($p < .05$)

Elements		N	Avg.	Std.	t	P
No.11 : Lecture grade	Pre	137	2.80	1.034	-5.217	.000
	Post	188	3.53	1.163		
No.13 : Hope to have additional software education	Pre	137	1.46	.500	4.634	.000
	Post	188	1.22	.418		

학생들의 학업 성적($t = -5.217$, $p = 0.000$), 추가 소프트

웨어 교육 희망 여부($t = 4.634$, $p = 0.000$)에서 유의미한 결과가 나타난 것으로 봐서 학생의 만족도와 함께 실제 학업 성적에서도 문제 해결 학습이 효과가 있음을 알 수 있다. 또한 문제해결학습을 수행했을 때 추가적인 소프트웨어교육을 희망하는 것을 알 수 있다.

따라서 소프트웨어 교육을 배우기 어려워하는 컴퓨터 비전공자들에게는 문제 해결 학습이 적절한 교육 방법이 될 수 있으며, 학생들에게 적합한 문제를 구성해서 생각해 볼 수 있는 시간을 주고, 스스로 문제를 해결해 나가는 과정이 교육 효과가 좋을 수 있다.

5. 결론

로봇과 인공지능으로 인한 커다란 사회적 변혁이 예상되는 미래에는 인문학적인 소양을 바탕으로 컴퓨팅 사고력과 문제 해결 능력이 뛰어난 인재를 필요로 할 것으로 예견되고 있다.

이에 본 논문에서는 대학에서 교육용 프로그래밍 언어로 가장 많이 사용되고 있는 Python을 바탕으로 소프트웨어 교육을 문제 해결 중심의 형태로 설계하고, 제시된 문제 상황에 맞는 해결 방법을 배우고 활용하여 스스로 해결해 나가도록 하였다. 그 결과 학생들의 지식 습득 정도와 학업 성적 향상 등에서 유의미한 결과를 얻었다.

향후에는 실제 학생들이 수행한 문제 해결 학습 과정과 출결, 학업 성적과의 상관관계를 분석하고, 이를 바탕으로 컴퓨팅 사고력과 문제해결력을 더욱 향상시킬 수 있는 방안에 대한 연구와 비전공 영역별로 실생활 문제의 적용 주제를 달리하여 소프트웨어 교육을 실시한 후 효과성을 검증하는 연구를 진행할 계획이다.

REFERENCES

- [1] Software Policy Division. (2015). Plan to educate human resources for software-oriented society, Ministry of Science and ICT. <http://www.msit.go.kr/web/msipContents/contentsView.do?cateId=mssw315&artId=1270998>
- [2] G. Chen, J. Shen, L. Barth-Cohen, S. Jiang, X. Huang & M. Eltouhky. (2017). Assessing Elementary Students' Computational Thinking in Everyday Reasoning and Robotics Programming. *Computer and Education*, 109,

- 162-175.
- [3] I. Jeong. (2017). Study on the Preliminary Teachers' Perception for the Development of Curriculum of the Robot-based Software Education in the Universities of Education. *Journal of The Korean Association of Information Education*, 21(3), 277-284.
- [4] Y. Jeon & T. Kim. (2015). The Design and Application of an Experience-Driven Online Software Class Based on Creative Problem Solving for Cultivating the Creative Personality of the Elementary Informatics-Gifted Students. *The Journal of Korea Elementary Education*, 26(4), 477-494.
- [5] S. Paik. (2017). *The Effects of Educational Programming Language with PBL(Problem Based Learning) on logical thinking ability and Problem Solving ability in elementary school environments*. Master thesis. Korea National University of Education, Chung-Buk.
- [6] H. J. Myung. (2014). *Effects of Software Special Classon Programming and Creative Problem Solving Capability*. Master thesis. Hanyang University, Seoul.
- [7] KERIS(Korea Education & Research Information Service). (2016). *Software education teacher raining program*. TM 2016-37.
- [8] B. Kim, Y. Jeon, J. Kim & T. Kim. (2016). Development and Application of Real Life Problem Solving Lesson Contents Based on Computational Thinking for Informatics Integrated-Gifted Elementary School Students' Creativity. *Korean Journal of Teacher Education*, 32(1), 159-186.
- [9] J. Ku, Y. Jeon & T. Kim. (2016). The Development and Application of Lesson Contents Based on the CT-CPS Framework for Improving the Creative Problem Solving Ability of Elementary Informatics Gifted Students. *The Journal of Korea Elementary Education*, 27(2), 339-357.
- [10] H. Y. Jung. (2014). An Empirical Study on Information Liberal Education in University based on IT Fluency and Computational Thinking Concept. *Journal of the Korea society of computer and information*, 19(2), 263-274.
- [11] K. Kim & H. Kim. (2014). A Case Study on Necessity of Computer Programming for Interdisciplinary Education, *Journal of Digital Convergence*, 12(11), 339-348.
- [12] J. Seo. (2017). A Case Study on Programming Learning of Non-SW Majors for SW Convergence Education. *Journal of Digital Convergence*, 15(7), 123-132.
- [13] S. H. Kim. (2015). Analysis of Non-Computer Majors' Difficulties in Computational Thinking Education. *The Journal of Korean Association of Computer Education*, 18(3), 15-23.
- [14] J. Y. Ki. (2018). A Study on UX Design Process Lecture Based on Modified PBL(Problem-Based Learning). *Journal of the Korea Convergence Society*, 9(1), 117-131.
- [15] J. H. Ku. (2017). Designing an App Inventor Curriculum for Computational Thinking based Non-majors Software Education. *Journal of Convergence for Information Technology*, 7(1), 61-66.

이 영 석(Lee, Youngseok)

[중신회원]



- 1999년 2월 : 서울교육대학교 초등교육과 (교육학사)
- 2001년 2월 : 서울교육대학교 컴퓨터교육과 (교육학석사)
- 2009년 8월 : 한양대학교 전자통신전파공학과 (공학박사)
- 2016년 3월 ~ 현재 : 강남대학교 KNU 참인재대학 컴퓨터 프로그래밍 교육 담당 교수
- 관심분야 : 소프트웨어 교육, 스마트러닝, 지능형 웹 정보 시스템
- E-Mail : yslee38@kangnam.ac.kr