Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

Application of Computational Thinking to Improve Learning Effectiveness in the Subject 'Programming'

Jae-Hyun Park[1], **Durk-Won Park**[*2]

[1]*Chuncheon Hansaem High School*
[2]*School of Computer science, Semyung University*

ABSTRACT

Computer education at school uses a lecture method in which instructors communicate ICT knowledge and software usage to learners. However, it has been pointed out that it is difficult to cultivate talented people with creative thinking and problem-solving skills necessary for the digital age. On the other hand, software education improves higher cognitive abilities such as learner 's logical thinking, problem solving ability, achievement and confidence through completion, cooperation and interaction through discussion. To apply the software education to the school site, it is necessary to provide software education centered on student activities in addition to the existing lecture class method. The purpose of this study is to apply computing thinking activity of software education to programming learning and to analyze its learning effect. To do this, we analyze the programming textbooks and design a class model to apply computing thinking activities to learning. And applied it to the actual class and compared it with the existing lecture method. In this study, we divided into experiment group applying computational thinking activities method and control group applying existing lecture class. Prior to the experiments, assured the homogeneity of two groups by academic achievement pre-test, after that, analyzed results of t-test by academic achievement post-test. As a result, the computational thinking activity applied to the programming lesson showed positive results both in interest and attitude of programming learning as well as achievement improvement.

KEYWORDS: Computational thinking, CS activity, Programming, Software education

ARTICLE INFO: Received 15 March 2018, Revised 8 April 2018, Accepted 13 April 2018.

[*]Corresponding author is with the School of Computer Science, Semyung University.

E-mail address: pdw403@semyung.ac.kr

1. 서 론

교육부는 2014년 7월 23일 '소프트웨어 중심사회 실현 전략 보고대회'에서 창조경제 시대에 필요한 창의적 사고력과 문제해결력을 갖춘 인재 양성 기반 조성을 위해서 학교에서 소프트웨어 교육을 체계적으로 배울 수 있는 교육 기회를 확대하고 우수 인재를 조기에 발굴 및 육성하겠다고 발표했다[1]. 그러나 지금까지 학교에서의 컴퓨터 교육은 교수자 주도로 ICT 관련 지식과 소프트웨어 사용법을 학습자에게 전달하는 강의식으로, 디지털 시대에 필요한 창의적 사고력과 문제해결력을 갖춘 인재를 양성하기 어렵다는 지적을 계속 받아왔다. 반면 그 대안으로 떠오르고 있는 소프트웨어 교육은 학습자의 논리적 사고력 증진, 문제해결 능력 신장, 완성을 통한 성취감과 자신감의 획득, 토론을 통한 협동심과 상호작용 효과와 같은 고등 인지 능력을 향상시킨다는 많은 연구결과가 발표되었다[2]. 이러한 소프트웨어 교육을 학교 현장에 적용하여 더욱 큰 효과를 보기 위해서는 앞서 언급한 강의식 수업 방법 이외에 학생 활동 중심의 소프트웨어 교육이 이루어져야 한다.

이에 본 연구에서는 소프트웨어교육의 학습 효과를 높일 수 있도록 프로그래밍 학습에 컴퓨팅 사고 활동을 적용하여 그 효과를 분석하였다. 이를 위한 논문의 구성은 다음과 같다. 제 2장에서는 컴퓨팅 사고에 관한 기존 연구를 설명하였다. 제 3장에서는 프로그래밍 교재를 분석하여 내용을 재구성한 후 이것을 기반으로 컴퓨팅 사고 활동을 수업에 적용할 수 있도록 설계한 수업 모델을 제시하였고, 제 4장에서는 이를 실제 수업에 적용하여 본 연구에서 제시한 컴퓨팅 사고 활동이 프로그래밍 학습의 흥미, 태도, 성취도 향상에 미치는 영향을 분석하였다. 마지막으로 제 5장에서는 결론 및 추후 제안사항을 제시하였다.

2. 관련 연구

2.1 컴퓨팅 사고

컴퓨팅 사고에 관한 연구는 미국 MIT의 S. Papert교수가 그의 저서에서 '절차적 사고'라는 용어를 사용한 이후, 2006년 미국 카네기멜론 대학의 Wing교수의 연구에 의해 'Computational Thinking'이라는 용어가 재조명되면서 컴퓨터교육 현장에서 이슈화되기 시작하였다[3,4]. 컴퓨팅 사고는 실생활에서 발생하는 복잡한 문제를 효과적이고 효율적으로 해결하기 위해 컴퓨팅의 기본적인 개념과 원리를 활용하는 인간의 사고과정 또는 종합적인 능력으로, 크게 추상화, 자동화 단계로 구성된다[5,6]. 추상화는 실생활의 문제를 컴퓨터를 통해 해결 가능한 형태로 표현하기 위한 사고과정으로 이를 위해 문제 해결에 필요한 자료를 수집 및 분석하고, 도표, 그래프 등을 활용하여 자료나 정보를 논리적으로 구조화하여 보기 쉽게 나타낸다. 자동화 단계는 추상화 단계에서 만들어진 알고리즘을 컴퓨터가 이해할 수 있도록 프로그래밍언어로 구현하고, 이를 실행함으로써 문제해결과정을 자동화하고 이와 같은 문제해결과정은 다른 문제에도 적용할 수 있도록 일반화한다.

2.2 기존 연구

전통적으로 소프트웨어 교육에서는 알고리즘과 프로그래밍에 대한 학습이 이루어지며 소프트웨어를 구현하기 위해서는 프로그래밍이 필수적이므로 교육기관에서는 프로그래밍 언어를 중심으로 한 교육이 이루어져왔다.

최형신은 컴퓨팅 사고의 세부역량을 정의하고 스크래치를 활용한 교육에서 컴퓨팅 사고에 대한 세부역량을 증진할 수 있는 수업내용을 제시하였으나, 구체적인 교수학습방법에 대한 연구는 진행되지 않았다[7]. 김수환, 한선관은 디자인 기반학습을 적용한 컴퓨팅 사고 교육의 효과로 컴퓨팅 사고 능력의 향상과 자기 프로그래밍 능력의 향상 등을 제시하

였으며, 디자인 기반학습이 컴퓨팅 사고 교육에서 효과적임을 밝혔다[8]. 특히, 교육단계에 따른 효과적 교육전략으로 친구들과 협력하기, 창작물 공유 및 발표하기, 다른 친구 작품과 결합하기 등을 제시하였다. 박정호의 경우는 이솝우화를 바탕으로 한 스토리텔링 기반의 소프트웨어 교육을 실시하고 소프트웨어 개념, 구현능력, 교육태도가 향상되었음을 검증하였다[9]. 또한 소프트웨어 교육 지침에 의하면 소프트웨어교육은 프로그램을 개발할 수 있는 역량을 교육하기 보다는 컴퓨팅 사고를 기를 수 있는 교육내용과 방법이 전개되어야 하고, 이를 위해서는 지식 위주의 교육보다는 수행 중심의 교육이 이루어질 수 있도록 교수학습방법과 평가방법을 제시할 필요가 있다고 강조하고 있다. 또한 문제해결 활동에 있어서 협력과 프로젝트 학습, 효과적인 의사소통을 포함한 학습 활동이 이루어지도록 설계하도록 권장하고 있다.

이상의 연구들에서 나타난 바와 같이 국내의 컴퓨팅 사고에 대한 교육은 교육내용에 대한 연구와 교육방법에 대한 연구가 시작되고 있는 시점이다.

3. 컴퓨팅 사고를 위한 교수·학습 방법의 실제

3.1 컴퓨팅 사고를 이용한 수업 설계

컴퓨팅 사고를 프로그래밍 학습에 적용시키고, 효과를 분석하기 위하여 먼저 관련 문헌 연구를 통해 컴퓨팅 사고의 세부 역량을 도출하였고, 선택된 컴퓨팅 사고 세부 역량을 다룰 수 있는 프로그래밍 수업 내용을 설계하기 위해 2007 개정 교육과정에 따라 출판된 2011년 정보 교과서를 이용하였다[10].

또한 국내·외의 프로그래밍 관련 연구들을 검토하여 흐름을 파악하고 컴퓨팅 사고 관련 평가에 초점을 두고 있는 연구들로 본 연구에서 설계하고자 하는 프로그램 설계 및 평가 방안에 관한 기초 자료를 확보한 후 이를 통해 프로그래밍 수업 내용을 개발하였다.

3.2 컴퓨팅 사고 활동 교수·학습 지도안 설계방향 및 개발

이 연구는 춘천에 위치한 특성화 고등학교 3학년 2개 반 54명을 대상으로 하였고, 학습 내용은 도출된 6개의 컴퓨팅 사고 활동을 진행할 수 있도록 프로그래밍 수업을 설계하였다.

본 연구에서 사용한 컴퓨팅 사고 세부 역량은 절차 및 알고리즘, 병행화 및 동기화, 자료 표현, 추상화, 문제 분해, 시뮬레이션의 6개 분야로 이는 모두 CSTA에서 제시한 요소로 시뮬레이션 요소만을 제외하고는 컴퓨팅 사고평가 프레임워크의 컴퓨팅 사고 요소와도 일치한다[11,12].

수업은 스탠포드 대학교에서 제시한 디자인사고 과정을 따르는 NDIS 모델을 이용하였다. 이 모델은 프로그래밍이 단지 기계적인 제품 개발을 위한 것이 아닌 인간의 삶을 개선하는 창의적 설계를 진행하는 것이라는 것을 학습자에게 인식시키며, 컴퓨팅 사고를 신장시키기 위한 설계와 개발의 과정을 통해 시뮬레이션을 제작하는데 유리한 모델이다.

표 1. NDIS 모델의 개요
Table 1. NDIS model

Process	Learning method
Needs	· 주어진 문제에 대한 고찰 · 사용자 중심 요구 분석
Design	· 분해와 패턴 찾기 · 알고리즘 설계
Implementation	· 프로그래밍과 피지컬 컴퓨팅을 이용한 산출물 구현
Share	· 산출물 공유 · 피드백을 통한 자기 성찰

또한 개발된 결과를 공유와 평가를 통해 개선의 방법을 찾는 선순환 구조를 갖는 모델로 개요는

<표 1>과 같다.

3.2.1 로봇청소기 알고리즘 구현

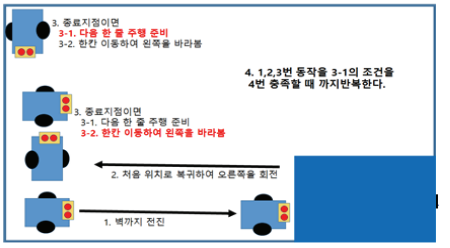
1차시에는 '로봇청소기 알고리즘 구현'을 주제로 하여, 학습자들이 NDIS 모델을 이용하여 로봇청소기의 기능을 분해하여 패턴을 찾고, 이를 프로그래밍과 피지컬 컴퓨팅을 이용하여 산출물을 구현하도록 구성하였으며 컴퓨팅 사고 요소로 문제분해, 패턴인식, 추상화, 알고리즘, 시뮬레이션을 사용하였다.

```
1  void gotoWall(){
2    while(getUltraSonic() > 10) {
3      leftMotor(MOTOR_SPEED);
4      rightMotor(MOTOR_SPEED);
5    }
6    move(0,0,DELAY_TIME);
7  }
8
9  void shuttle(){
10   gotoWall( );
11   move(-MOTOR_SPEED,MOTOR_SPEED,TURN_TIME_180);
12   move(0,0,DELAY_TIME);
13   gotoWall( );
14   move(MOTOR_SPEED,-MOTOR_SPEED,TURN_TIME_90);
15   move(0,0,DELAY_TIME);
16 }
17
18 void loop(){
19   int count=0;
20   while(count<4){
21     shuttle();
22     if(getUltraSonic() < 10){
23       move(MOTOR_SPEED,MOTOR_SPEED,TURN_TIME_180);
24       move(0,0,DELAY_TIME);
25       move(-MOTOR_SPEED,MOTOR_SPEED,TURN_TIME_180);
26       move(0,0,DELAY_TIME);
27       count++;
28     }
29     else{
30       move(MOTOR_SPEED,-MOTOR_SPEED,TURN_TIME_90);
31       move(0,0,DELAY_TIME);
32     }
33 } }
```

그림 1. 프로그램 소스 코드
Figure 1. Source code

이를 통해 개발한 수업지도안은 <표 2>로 설계 방향에 맞추어 학습자들이 주어진 문제를 해결할 수 있도록 하였고, 컴퓨팅 사고 활동 중 실제 구현한 프로그램의 소스코드는 <그림 1>과 같다.

표 2. NDIS 모델을 적용한 수업지도안
Table 2. Lesson plan applying NDIS model

Learning Phase	Teaching and Learning Activities
Needs	· 상황 제시를 통하여 청소로봇의 필요성을 이해한다. · 일반적인 로봇 청소기가 수행해야할 기능이 무엇인지를 도출한다. 예) 먼지제거, 자동이동, 충전 등 · 로봇 청소기의 기능 중 만들고자 하는 로봇의 기능에 대하여 정의한다. 예) 어떤 방식으로 이동하여 청소할 것인가
Design	· 패턴인식 : 전체는 한 칸 이동을 전체만큼 반복하는 패턴을 찾아내도록 지도한다. · 추상화 : 바둑판처럼 되어 있는 바닥을 청소하며 로봇이 움직일 때는 한 칸씩 움직이도록 단순화 한다. · 알고리즘 : 한 칸씩 이동, 벽을 만나면 회전하여 처음 위치로 복귀하고 계속 이동한다. 
Implement-ion	· 설계된 알고리즘을 바탕으로 프로그램으로 구현하여 알고리즘대로 로봇이 이동하는지 확인한다. · 프로그램을 실행해보고 오류사항을 발견하게 되면 디버깅을 수행하도록 지도한다.
Share	· 모둠별로 완성된 프로그램을 발표하고 공유해본다. · 좋은 점과 개선할 점을 이야기하고, 여기서 나온 피드백을 중심으로 개선점을 반영하여 구현해 본다. · 개발 과정에 대한 간략한 소개 글을 작성하여 온라인에 탑재하고 댓글 등의 반응을 통해 자기성찰을 한다. · 개발과정에서의 자기 성찰을 해보고 몇몇 학생들에게 발표하도록 한다.

3.2.2 커튼 및 조명 자동화 알고리즘 구현

2차시에는 햇빛의 밝기에 따른 커튼 작동 및 조명 작동에 대한 관계를 이해하여 자동화에 필요한 요소를 분해하여 패턴을 찾아 시각화하고 알고리즘을 설계하여 프로그래밍 하는 전체 과정을 이해할 수 있도록 <표 3>과 같이 수업지도안을 개발하였다.

표 3. NDIS 모델을 적용한 수업지도안
Table 3. Lesson plan applying NDIS model

Learning Phase	Teaching and Learning Activities
Needs	· 상황 제시를 통하여 커튼 및 조명 자동화의 필요성을 이해한다. · 자동화가 인간에게 얼마나 이로움을 가져다 주는지 그 가치를 탐색한다. · 자동화를 통해 구해야 할 것이 무엇인지 이해하고 분석한 후 탐색한 내용을 모둠별로 발표한다. [활동] 문제를 분석하여 정리해 보자. 문제 상황의 이해 / 문제 해결 방법 만들기 / 필요한 자료의 수집 해결할 문제의 최종목표는 무엇인지 생각해 보자. / 어떻게 문제를 해결할 수 있을지 아이디어를 모으고 해결 방법을 창의적으로 만들어 보자. / 문제 해결에 필요한 자료는 어떤 것이 있을지 찾아보자. 아이디어: 방법:
Design	· 문제분해 : 빛과 커튼, 조명과의 상호 동작 관계를 파악한 후, 빛의 밝기에 따라 처리 가능한 작은 단위의 문제로 분해하여 필요한 요소를 추출하여 구조화 해 본다. · 패턴인식 : 요소 간 패턴을 쉽게 파악할 수 있도록 시각화하며, 이 때 마인드 맵, 브레인스토밍, 그래프와 도식화 등의 다양한 전략을 사용하도록 지도한다. · 추상화 : 빛의 밝기에 따라 변화되는 커튼과 조명의 동작 패턴을 공식으로 표현한다. · 알고리즘 : 구현할 알고리즘을 설계하여 순서도, 의사코드 등으로 표현한다.
Implement-ion	· 센서나 기타 필요한 교구를 이용하여 실제 상황과 비슷한 환경을 만들어 제작한다. · 실제로 커튼과 조명을 자동으로 조절하는 프로그램을 구현한다.
Share	· 프로그램을 발표하고 공유한다. · 다른 그룹의 작품과 비교하여 자신들의 프로그램을 수정한다. · 완성된 작품을 공유하고 자기 성찰을 한다.

4. 컴퓨팅 사고 활동을 적용한 교수·학습 방법의 효과성 검증

4.1 연구의 설계

이 연구에서는 2017년 9월부터 10월까지 프로그래밍 수업에 컴퓨팅 사고 활동을 이용하여 학습한 후 결과를 분석하였다. 연구대상의 구성은 강원도 춘천 소재의 특성화 고등학교 3학년 학생 46명을 실험집단 20명과 통제집단 26명으로 나누어 적용하였다. 실험집단은 컴퓨팅 사고 활동을 이용하여 학습을 진행하였고, 통제집단은 기존의 전통적 방식으로 일반적인 프로그래밍 학습을 진행하여 컴퓨팅 사고 활동을 활용한 프로그래밍 교과 수업이 프로그래밍에 대한 흥미, 태도 및 성취도에 미치는 영향을 검증하였다.

4.2 검사 도구 및 분석 방법

결과 분석에 사용한 검사 측정도구는 학업 흥미도, 학업 태도, 성취도 검사이다.

학업 흥미는 프로그래밍 학습에 대한 학생 흥미도를 조사한 것으로, 전공자 3인을 통해 타당성을 검증받은 4문항을 사용하였다.

기존 다수의 연구에서 Likert 3점을 활용함에 따라, 본 연구의 도구는 모두 Likert 3점 척도로 연구

를 진행하였으며 문항내적일치도 신뢰도 계수인 Cronbach의 α는 .75로 신뢰성을 확보하였다[13].

학업 태도는 프로그래밍 구현에 대한 태도, 프로그래밍 태도의 적용에 대한 문항을 Likert 3점 척도의 6문항을 전공자 3인을 통해 타당성을 검증받아 사용하였으며 문항내적일치도 신뢰도 계수인 Cronbach의 α는 .82로 신뢰성을 확보하였다.

본 연구는 프로그래밍 학습에 컴퓨팅 사고 활동을 적용한 집단과 기존 방식으로 학습한 집단의 프로그래밍에 대한 흥미, 태도 및 성취도의 차이를 분석하기 위해 SPSS를 활용하였다.

4.3 학습 효과성 분석

4.3.1 프로그래밍 교과 학업흥미

컴퓨팅 사고 활동과 기존 방식을 이용한 프로그래밍 학습 성과인 프로그래밍 흥미에 대한 비교결과는 <표 4>와 같다.

표 4. 학업태도 하위요소별 집단 간 사후 검사 결과
Table 4. Result of post-test for academic attitudes element

Element	Group	M	SD	MD	F
관심	실험집단	2.80	.41	.72	.13*
	통제집단	2.08	.48		
재미	실험집단	2.80	.41	.88	12.85*
	통제집단	1.92	.85		
흥미	실험집단	2.50	.51	.42	.04*
	통제집단	2.08	.67		
타 교과 비교	실험집단	2.50	.68	.65	.09*
	통제집단	1.85	.62		
프로그래밍흥미 전체	실험집단	2.65	.38	.67	15.27*
	통제집단	1.98	.42		

* $p < .05$

컴퓨팅 사고 활동 집단이 기존 학습 방법 집단에 비해 프로그래밍에 대한 흥미 전체 영역의 평균이 높게 나타났다. <표 4>에서 보면 컴퓨팅 활동 집단은 관심, 재미, 흥미, 타 교과 비교순으로 프로그래밍에 대한 흥미가 높게 나타났으나, 기존 방식으로 학습한 집단에서는 관심, 흥미, 재미, 타 교과 비교순으로 나타났다. 영역별 차이는 재미, 관심, 타 교과 비교, 흥미 순이다. 두 집단 간의 모든 프로그래밍 흥미 영역별 평균은 유의수준 5%에서 통계적으로 유의한 차이가 나타났다.

4.3.2 프로그래밍 교과 학업 태도

컴퓨팅 사고 활동 집단과 기존 방식 학습 집단의 프로그래밍 학습 성과인 프로그래밍 태도에 대해 비교한 결과는 <표 5>와 같다.

표 5. 학업태도 하위요소별 집단 간 사후 검사 결과
Table 5. Result of post-test for academic attitudes element

Element	Group	M	SD	MD	F
태도의 적용	실험집단	2.90	.30	.21	15.27
	통제집단	2.69	.47		
구현에 대한 태도	실험집단	2.80	.41	.34	13.92
	통제집단	2.46	.76		
프로그래밍태도 전체	실험집단	2.85	.23	.28	18.56
	통제집단	2.57	.52		

* $p < .05$

컴퓨팅 사고 활동 집단이 기존 방식으로 학습한 집단에 비해 프로그래밍 태도 전체 영역의 평균이 높게 나타났다. 두 집단 모두 태도의 적용, 구현에 대한 태도 순으로 프로그래밍 태도가 높게 나타났다. 영역별 차이는 구현에 대한 태도, 태도의 적용 순이다. 두 집단 간의 모든 프로그래밍 학습 태도 영역별 평균은 유의수준 5%에서 통계적으로 유의한 차이가 나타나지 않았다.

4.3.3 컴퓨팅 사고 활동 성취도

컴퓨팅 사고 활동 집단과 기존 방식 학습 집단의 프로그래밍 학습의 성과인 성취도에 대한 비교결과는 <표 6>과 같다. 컴퓨팅 사고 활동 집단의 평균은 1.68, 기존 방식 학습 집단의 평균은 .87로 기존 방식 집단에 비해 성취도가 유의하게 높게 나타났다.

표 6. 학업태도 하위요소별 집단 간 사후 검사 결과
Table 6. Result of post-test for academic attitudes element

Element	Group	M	SD	MD	F
성취도 전체	실험집단	1.68	.53	.81	6.63*
	통제집단	.87	.83		

* $p < .05$

5. 결론 및 제안사항

본 연구는 컴퓨팅 사고 활동을 활용한 프로그래밍 학습이 기존의 프로그래밍 학습과 비교하여 교육성과인 프로그래밍 흥미, 프로그래밍 태도 및 성취도에 어떠한 영향을 미치는지 알아보고자 하였다. 본 연구의 결과는 다음과 같다.

첫째, 본 연구는 기존 방식 학습 집단보다 컴퓨팅 사고 활동을 활용한 집단의 프로그래밍에 대한 흥미도가 더 높게 나타났다. 이는 다양한 상황을 제시하고 이를 해결하기 위한 컴퓨팅 사고를 이용하여 프로그래밍에 대한 흥미를 높이는 것으로 ~의 연구와 동일한 결과이다. 또한 학습자는 생활이나 자신의 관심분야와 관련된 되거나 직접 참여할 수 있는 경우에 학습흥미가 높게 나타난다는 연구를 뒷받침한다[14]. 본 연구에서 로봇 청소기 알고리즘 구현 및 햇빛과 커튼, 조명과의 관계를 분석하여 시뮬레이션 하는 컴퓨팅 사고 활동을 제공하였기에 기존 학습 집단 보다 프로그래밍 흥미 향상에 유의한 영향을 미쳤음을 알 수 있다. 따라서 프로그래밍 학습을 설계하는 담당자들은 실제 생활에 접할 수 있는 다양한 학습활동을 제공해야 할 필요가 있음을 시사한다.

둘째 성취도는 기존 학습 집단에 비해 컴퓨팅 사고 활동 집단이 더 높게 나타났다. 이는 프로그래밍을 사용하여 주어진 상황을 해결하는 컴퓨팅 사고 활동이 학습자의 정보 이해력을 향상시켜 과제 해결 능력에 영향을 미쳤기 때문으로 보인다. 또한 정보의 적시성[15]은 자신의 학습내용과 관련성이 높은 곳에서 문제를 해결할 수 있게 도왔기에 성취도를 높이는데 기인한 것으로 보인다. 그러나 평균이 낮게 나타난 이유는 전문적인 프로그래밍 용어의 이해 부족의 영향으로 보이므로, 프로그래밍 학습을 설계하는 담당자들은 학습 대상자의 교육과정 수준에 맞춘 컴퓨팅 사고 활동지를 개발하여야 할 것이다.

셋째 컴퓨팅 사고 활동 집단은 기존 학습 집단보다 프로그래밍에 대한 태도가 높게 나타났으나, 유의한 결과는 아니었다. 이는 프로그래밍에 대한 태도는 온라인 져지 및 프로젝트 수행 등 다양한 문제 해결 과정을 직접 경험할 때 증진되기 때문에 컴퓨팅 사고 활동 자체로는 유의하게 나타나지 않은 것으로 보인다. 또한 학습을 위한 시간 및 프로그래밍 프로젝트 수행과 연결성이 부족하여 프로그래밍에 대한 태도가 유의하지 않은 결과를 나타낸 것으로 파악된다. 프로그래밍의 경우에는 학교수업에서는 제공할 수 없는 실제 프로젝트 진행 과정이나 전문적 지식을 이용한 직접적인 체험과 구현의 기회를 제공하여 동기를 부여하고 다양한 교육적 기회를 제공할 수 있도록 학습 모형을 설계해야만 할 것이다.

향후 연구과제로는 본 연구에서 적용한 수업 모델 외에도 위에서 제시한 학습 설계 방향을 활용한 지도안이 더 개발되어 학교 현장에서 프로그래밍 학습에 컴퓨팅 사고 활동을 활용한 수업이 적

극적으로 개발되어야 할 것이다.

References

[1] Ministry of education, *Software-oriented social realization strategy report meeting*, Press Release, pp. 1-2, Jul. 2014.

[2] Y. J. Jeon, *The development and aplication of a CT-CPS(Computational Thinking-based Creative Problem Solving) instructional model for the software education of new curiculum*, Korea National University of Education Graduate School Doctoral Thesis, pp. 87-96, Feb. 2017.

[3] S. A. Papert, *Mindstorms: Children, computers, and powerful ideas*, Cambridge, MA: Perseus Publishing, pp. 179-203, 1993.

[4] J. M. Wing, *Computational thinking, Communications of the ACM,* Vol. 19, No. 3, pp. 33-35, 2006.

[5] M. Y. Choi, *Development of a convergent talent education program for improving elementary school students' thinking ability of computing*, Gyeongin National University of Education Master's Thesis, 2015.

[6] T. H. Kim, *Programming-oriented STEAM training program for improving computing thinking*, Jeju University Master's Thesis, 2015.

[7] H. S. Choi, *Developing lessons and rubrics to promote computational thinking*, Journal of The Korean Association of information Education, Vol. 18, No. 1, Feb. 2014.

[8] S. H. Kim, and S. K. Han, *Design-based learning for computational thinking*, Journal of The Korean Association of information Education, Vol. 16, No. 3, Jul. 2012.

[9] J. H. Park, *A study on digital storytelling based programming education*, Journal of The Korean Association of information Education, Vol. 19, No. 5, PP. 119-128, 2014.

[10] W. G. Lee, S. Y. Jung, and S. W. Yoo, *(highschool) Information : Teacher's guide*, MiraeNCulture, pp. 148-152, Mar. 2011.

[11] Computer Science Teachers Association, Computational Thinking, http://csta.acm.org/Curriculum/sub/CompThinking.html, Oct. 2017.

[12] K. Brennan, and M. Resnick, *New frameworks for studying and assessing the development of computational thinking*, Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada, 2012.

[13] G. O. Lee, and N. Y. Jung, *Effect of plant experience activity on children's emotional development,* Journal of Korean practical arts education, Vol. 21, No. 1, pp. 113-128, 2008.

[14] H. S. Jang, and K. H. Choi, *The effects of science museum field trips on middle school students' awareness about S-T-S interactions*, The Journal of the Learner-centered curriculum education society, Vol. 6, No. 2, pp. 425-445, 2006.

[15] M. S. Donovan, J. D. Bransford, and J. W. Pellegrino, *How people learn*, https://www.researchgate.net/figure/Bransford-et-al-2000-How-People-Learn_fig1_263809146, Aug. 2017.

프로그래밍 학습효과 향상을 위한 컴퓨팅 사고 활용 방안 연구

박재현[1], 박덕원[2]
[1]*춘천한샘고등학교*
[2]*세명대학교 컴퓨터학부*

요 약

학교에서의 컴퓨터 교육은 교수자 주도로 ICT 관련 지식과 소프트웨어 사용법을 학습자에게 전달하는 강의식으로, 디지털시대에 필요한 창의적 사고력과 문제해결력을 갖춘 인재를 양성하기 어렵다는 지적을 계속 받아왔다. 반면 그 대안으로 떠오르고 있는 소프트웨어 교육은 학습자의 논리적 사고력 증진, 문제해결 능력 신장, 완성을 통한 성취감과 자신감의 획득, 토론을 통한 협동심과 상호작용 효과와 같은 고등 인지 능력을 향상시킨다. 이러한 소프트웨어 교육을 학교 현장에 적용하여 더욱 큰 효과를 보기 위해서는 기존의 강의식 수업 방법 이외에 학생 활동 중심의 소프트웨어 교육이 이루어져야 한다. 이에 본 연구에서는 소프트웨어교육의 학습 효과를 높일 수 있도록 프로그래밍 학습에 컴퓨팅 사고 활동을 적용하여 그 효과를 분석하였다. 이를 위해 프로그래밍 교재를 분석하여 컴퓨팅 사고 활동을 수업에 적용할 수 있도록 수업 모델을 설계한 후 이를 실제 수업에 적용하여 기존의 전통적 강의방법과 비교하는 것으로 진행되었다. 연구대상은 특성화 고등학교 3학년 학생 54명을 대상으로 실험집단과 통제집단으로 나누어 프로그래밍 학습에서 컴퓨팅 사고 활동이 가져오는 긍정적인 요인들에 대해서 연구하였다. 그 결과 프로그래밍 교과 수업에 적용한 컴퓨팅 사고 활동은 프로그래밍 학습의 흥미, 태도뿐만 아니라 성취도 향상에서도 모두 긍정적인 결과를 보였다.

감사의 글

이 논문은 2017년도 세명대학교 교내 학술연구비 지원에 의해 수행된 연구임.

Jae Hyun Park received the B.S. and M.S. degree in the Department of Computer Science from the Semyung University in 1993 and 2001. He completed his Ph.D. in the Department of Computer Science from the Semyung University in 2015. He has been working in the Chuncheon Hansaem High School.

E-mail address: teacher.pjh@gmail.com

Durk Won Park received the B.S. and M.S. degree in the Department of Computer Science from the Soongsil university in 1982 and 1988. He completed his Ph.D. in the Department of Computer Science from the Chungnam University in 1997. He worked as a Professor in Department of Computer Science, Daeduk College from 1988 to 1991. He has been a processor in School of Computer Science, Semyung University since 1991. His current research interests include parallel processing and image processing.

E-mail address: pdw403@semyung.ac.kr