

초등학생들의 소프트웨어 교육을 위한 아이돌 교수 학습 모델

김갑수

서울교육대학교 컴퓨터교육과

요 약

제 4차 산업혁명에서 모든 기기들과 사람들이 상호 연동되어 새로운 부가가치를 창조하는 시대이다. 무엇보다도 중요한 것이 소프트웨어이기 때문에 초등학교에서 소프트웨어 교육이 필요하다. 그러나 지금까지 초등학교에서 소프트웨어를 교육하기 위한 교수 학습 방법이 부족하다. 본 연구에서는 이런 문제를 해결하기 위하여 소프트웨어 교육을 위한 IDOL 모델을 제안한다. 본 모형은 구현하기(implement), 다변화 시키기(diversify) 관찰하기(observe)를 통해서 학습하는(learning)것이다. 구현하기 단계에서는 초등학생들은 빨리 만들어 보는 것에 초점을 두었고, 다변화시키기 단계에서는 만든 것을 다양하게 변경하여 보는 것이고, 관찰하기 단계에서 변경된 것이 어떻게 다른지 관찰하고 평가하는 단계이다. 본 모형은 전문가 집단을 통한 델파이 검증한 결과 타당하다고 판별되었다.

키워드 : 소프트웨어 교육, 수업모형, 초등학생, 구현하기, 다변화시키기, 관찰하기

An IDOL teaching and learning model for software education of elementary students

Kapsu Kim

Dept. of Computer Education, Seoul National University of education

ABSTRACT

In the 4th Industrial Revolution, it is a time when all the devices and people interact to create new added value. Software education is necessary in elementary school because what is more important is software. However, until now there is a lack of teaching and learning methods for software education in elementary school. In this paper, IDOL model is proposed as a method to solve this problem. statements for elementary students. The idol model is learning through observe, to implement, to diversify. In the implementation, the elementary students are to make it quickly, In the diversification, it is variously changing the things that have been made, and observation how the changed things are different. A model is verified by a group of experts consisting of elementary school teachers.

Keywords : Software Education, Instructional Model, Elementary School, Implementation, Diversification, Observation

본 논문은 2018년도 서울교육대학교 교내연구비에 의하여 연구되었음

논문투고 : 2018-12-13

논문심사 : 2018-12-21

심사완료 : 2018-12-24

1. 서론

4차 산업 혁명에서는 수십조개의 센서들이 상화 연동되어 새로운 세상을 만들고 있고, 그 핵심에 소프트웨어가 있다는 것을 모두들 인식하고 있다. 따라서 세계 여러 나라는 소프트웨어 교육을 정규 교육과정에 편성하여 모든 학생들을 초등학교부터 소프트웨어 교육을 받으며 컴퓨팅 사고를 키워갈 수 있도록 하고 있다.

영국에서는 컴퓨팅 교육에 대해서는 초등학교 1학년부터 체계적으로 교육시키고[1,2]있다. 또한 각 단계로 체계적으로 성취기준과 교과서 등을 만들어 배포하고 있다.

미국에서도 ACM의 CSTA[5,6]에서도 유치원부터 체계적으로 초등학교에서 컴퓨팅 교육과정을 만들어 각 주별의 학군별로 교육과정을 전파하고 있다. 물론 컴퓨팅 교육과정을 운영하고 있지 않는 곳도 많이 있다. 또한 ISTE[3,4,11]에서도 ACM의 CSTA와 다르게 교육공학적인 관점에서 컴퓨터 기술을 이용하는 교육과정에 활용한 교육과정을 만들어서 이용하고 있다.

우리나라에서는 21세기의 정보화 시대를 준비하면서 2000년 12월에 초등학교 1학년부터 주 1시간씩 정보통신 기술교육[14]을 교육부의 지침으로 실시하였고, 이 지침을 2005년 12월에 개정하여 새로운 정보통신기술교육 지침을 만들어서 알고리즘과 정보 처리 능력을 추가하였다[15]. 그러나 2008년 말에 정보통신기술 교육지침이 폐지되었고 이후에 초등학교 컴퓨팅 교육은 점점 쇠퇴하기 시작하였다. 그 결과가 우리나라는 PISA의 통계자료 등에서 초등학생들의 컴퓨터 능력이 매우 저하되었다는 것을 알 수 있다[7,8,10]. 다행히 예비 교사들의 PISA의 데이터[9] 분석과 비교하여 보면 어느 정도 컴퓨팅 교육을 시킬 수 있는 자격은 되어 있는 것 같다. 2015년 개정 교육과정을 만들어 소프트웨어 교육 시간을 초등학교에서 성취기준 5개를 만들어 17시간으로 교육할 수 있도록 하였다[16,17].

2015 개정교육과정에서 프로그래밍을 구체적으로 학교 현장에서 실시하는 소프트웨어 교수 학습 방법이 필요하다. 지금까지 교수학습 모델은 소프트웨어 교육의 세부적인 내용보다 글로벌한 내용으로 구성되어 있어 있는 단점이 있다. 본 연구에서 학생들이 실제로 작은 프로그램을 직접 만들어 볼 수 있는 모델을 제안한다.

본 연구에서는 소프트웨어 교육을 바로 실시할 수 있

는 교수학습 모델인 IDOL 모델을 제안한다. 아이돌 모형은 구현하기(implement), 다변화시키기(diversify) 관찰하기(observe)를 통해서 학습하는(learning)것이다. 구현하기 단계에서는 초등학생들은 결과를 빨리 보는 것이 중요하기 때문에 간단히 만들어 보는 것에 초점을 두었다. 다변화시키기 단계에서는 간단히 만들어 본 것을 다양하게 학생들의 아이디어를 이용하여 변경하게 해 보는 것이다. 관찰하기 단계에서 변경한 것이 어떻게 다른지 관찰하고 평가하면서 서로 공유하는 단계이다. 초등학교 소프트웨어 교육은 3단계를 통해서 학습하는 것이다.

2장에서는 관련 연구들을 기술하고,3장에서는 제안한 소프트웨어 교육 모델을 설명하고, 4장에서는 제안한 모델을 검증하고, 5장은 결론이다.

2. 관련 연구

2.1 백워드 설계 모형

이영호, 구덕희(2015)[12]는 소프트웨어 교육 모형으로 백워드 설계 모형을 제안하였다. 이 모형은 결과 확인하기 단계, 다양한 이해의 증거 결정하기 단계, 학습경험과 수업 계획하기 단계로 구성되어 있다. 1단계에서는 “단원 수업을 통해 학생들에게 기하는 학습 결과를 명확히 규정하는 단계이다. 이 단계는 추출 된 일반화, 핵심 질문, 사실과 개념 지식, 기능/과정 지식 요소를 포함한다.”로 되어 있다. 2단계에서는 수행과제를 제시하고 평가하는 것이고, 3단계에서는 학습활동을 제안하였다.

백워드 설계 모형의 장점은 “소프트웨어적 지식/기능과 실제 현실을 연결하고, 소프트웨어 교육 운영 지침의 내용을 보다 유의미하고 효과적인 수업으로 구성할 수 있고, 소프트웨어 교과가 갖는 궁극적 목적과 단원별 목표 혹은 수업 단위 학습 목표를 유기적으로 연결할 수 있다”[5]고 주장하고 있다.

이 모형은 소프트웨어 교육과정을 결과를 먼저 예상하고 그 목표를 찾아가는 관점에서는 즉 수렴적 사고에서는 좋은 관점이지만 발산적인 사고를 위한 것에 부족한 점이 있다.

2.2. 스마트 기반 협력학습 모형

성영훈(2015)[18]은 초등학생의 프로그래밍 교육을 위해 스크래치를 활용한 스마트 기반 협력학습 모형을 설계하였다. 스마트 기반 협력 학습 모형은 스마트 리닝 도구를 학습 단계별로 이용하여 학생들이 학습에 어려운 점을 협력하여 해결하는 것이다. 성영훈(2015)은 스마트 기반 협력학습 모형은 개념 알기 단계, 구현하기 단계, 알아가기 단계와 공유하기 단계로 각 단계마다 학습 도구를 제안한 것이 큰 특징이고, 활동마다 협력하는 것이 주요 요소인 것 같다. 초등학생들은 학습을 할 때에 직관적인 학습하기 때문에 개념알기 단계에서 학생들이 지루하기 느낄 수 있는 단점이 있다.

2.3 컴퓨팅 사고력 기반 창의적 문제해결 (CT-CPS) 수업 모형

전용주(2017)는 소프트웨어 및 컴퓨팅에 관련된 사고 과정과 원리를 실생활의 소재와 관련지어 창의적이고 능동적으로 그 해결방안을 구현해가는 과정으로 제시할 수 있는 CT-CPS 수업 모형을 개발하였다[13].

인식 및 분석단계에는 동기유발 및 수업에서 활용할 SW 도구 선택하기, 문제 인식하기(발견하기)여 시각화 문제화하기로 구성되어 있다. 아이디어 구상 단계에는 소프트웨어를 통해 문제를 해결할 수 있는 아이디어를 다양하게 생각하기, 해결 아이디어를 단순화하기, 해결 아이디어의 설계하기, 시각과 논리를 종합한 전체 스토리보드 작성하기로 구성되어 있다. 구현 및 평가 단계에는 발표 및 시연하기(SW 및 PPT)와 평가/피드백 나누기로 구성되어 있다.

CT-CPS 수업모형 기반의 소프트웨어 수업이 창의적 문제해결력, 메타인지, 컴퓨터 학습태도, 학습동기 등 인지적, 정서적 영역에서 효과성이 있다고 주장하였다.

3. IDOL 수업 모형

3.1 IDOL 수업모형의 단계 및 구조도

IDOL 수업 모형은 초등학생들이 일상생활에서 컴퓨

터 게임이나 SNS 등을 많이 하면서 빨리 결과를 볼 수 있는 데 익숙하다. 따라서 소프트웨어 교육도 학생들이 빨리 만들어서 경험해 보는 것이 주요하기 때문에 제일 먼저 구현하는 것에 초점을 두어 학생들이 만든 것을 다양하게 변화하는 보면서 각종 기능을 익히거나 사고를 할 수 있고 학생들이 다양하게 변화시킨 것들을 평가하고 공유하여 정리하는 단계이다. 각 단계별로 교사가 할 일과 학생이 할 일을 정리하면 아래와 같다.

구현하기 단계에서는 교사할 일은 학습문제 및 예시 자료 제시하고, 관련 활동을 안내하고, 처음 만나는 명령어(블록) 설명 및 시범 보이기를 한다. 학생이 할 일은 학습 문제를 이해하고, 데이터를 찾고, 알고리즘을 설계하고, 구현하고, 테스트한다.

다변화(diversify)시키기 단계에서는 교사가 할 일은 학생들에게 다양하게 변경할 아이디어를 제공하고, 결과물을 공유하여 토론하게 한다. 학생이 할 일은 데이터, 알고리즘, 입출력이 변경된 것을 평가하고 공유하여 토론한다. 학생이 할 일은 데이터들을 변화시켜 보고, 알고리즘들을 변경하여 보고, 입력과 출력을 변경하여 본다.

관찰하기(observe) 단계에서 교사가 할 일은 관찰한 결과물 확인 및 피드백한다.

3.2 구현하기 단계

구현하기 단계에서는 학생들이 문제해결하기 위한 절차를 다음과 같은 세부 단계로 나눈다.

3.2.1 문제 이해(Understanding)하기

구현하기 단계에서는 첫 번째 단계로서 학생들이 문제를 처음에 이해할 수 있는 이해하기 단계이다. 이 단계에서는 학생들의 주어진 문제를 이해할 수 있어야 한다. 이해하기 위해서 오프라인 활동을 통하여 학생들이 세부적으로 문제를 나누어 문제들을 이해하여야 한다. 문제 이해를 위해서 컴퓨터가 처리하기 위한 방법으로 데이터 입력과 출력의 관계를 이해하여야 하고, 입력과 출력간의 관계를 찾아내는 것이 알고리즘이 된다. 이와 같이 문제를 이해한 후에 문제에 필요한 데이터를 찾는 단계로 넘어 간다.

다음 예제와 같은 문제1을 생각할 수 있다.

20세미만이면 3000원짜리 티켓을 20%할인해 주는 프로그램을 만들어 보자.

3.2.2 입출력 데이터 찾기

문제가 무엇인지 정확하게 이해한 후에 이 문제에 대한 필요한 데이터를 찾아낸다. 데이터는 입력 데이터와 출력 데이터를 찾아내어야 한다. 또한 이 데이터의 특성들을 이야기 하게 한다.

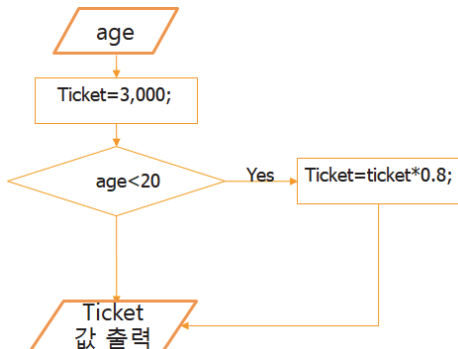
데이터의 특성은 숫자와 문자 데이터로 구성되어 있다는 것을 설명하고 데이터의 특성에 따라 연산이 다르다는 것을 학습해야 한다.

문제1에서는 입력값은 나이이고 출력값은 티켓값이다. 나이 데이터는 정수이라는 것이고, 출력값이 티켓값도 정수하는 것을 주의하게 한다.

3.2.3 알고리즘 설계하기

입력 데이터에서 출력 데이터로 변환하는 알고리즘을 설계하여야 한다. 알고리즘은 문제해결하기 위한 어떤 동작 (즉 명령어)들의 집합이다. 알고리즘은 알고리즘을 설계할 때에 가상코드로 설계하는 방법과 흐름도로 표현하는 방법이 있다.

문제1을 흐름도로 표현할 수 있고, 표현하면 다음 그림 [Figure 1]과 같다.



[Figure 1] Flowchart

문제1을 다음과 같은 가상코드로 표현할 수 있다

어떤 것을 클릭하면 동작하게 한다. [제어블럭]
 “나이를 입력하세요”를 스프라이트가 이야기 하고 답을 입력할 때까지 기다린다.[관찰블럭]

나이 변수에 입력한 값을 저장된다.[데이터블럭, 관찰블럭]

티켓변수에 3000원이 저장된다.[변수블럭]

나이 변수가 20보다 작으면 티켓값을 3000*0.8인 2400원으로 변경한다.[제어 블럭, 연산블럭, 데이터 블럭][데이터 블럭, 연산블럭, 데이터 블럭]

티켓값을 10초 동안스프라이트가 이야기 하게 한다.[형태 블럭, 연산블럭, 데이터블럭]

3.2.4 코딩하기

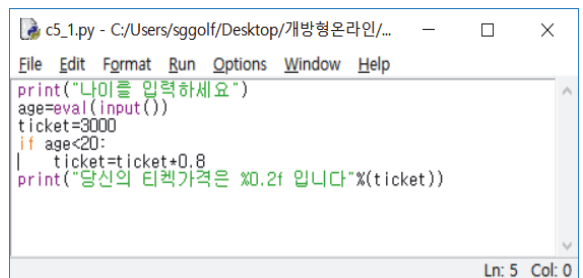
4단계에서 설계한 것을 개발하고자 하는 프로그래밍 언어를 사용하여 구현하여야 한다. 코딩은 일반적으로 많이 사용하고 있는 스크래치 블록 코딩으로 할 수 있거나 파이썬과 같은 텍스트 기반 언어로 코딩할 수 있다.

문제1을 스크래치로 표현하면 다음 [Figure 2]와 같다.



[Figure 2] Scratch coding

또한 문제1을 텍스트 코드인 파이썬으로 표현하면 다음 [Figure 3]과 같다. [Figure 3]은 알고리즘을 표현한



[Figure 3] Python coding

것을 파이썬 명령어로 그대로 표현한 것이다.

3.2.5 테스트하기

코딩한 결과를 원하는 결과가 나오는지 확인하는 것이 테스트하기 이다. 이때에 논리적인 에러 등을 찾아 발견한다.

테스트하기 위해서 테스트 데이터를 생성하는 것이 매우 중요하다. 일반적으로 테스트 데이터는 무한하기 때문에 좋은 테스트 데이터를 주는 것이 좋다.

조건문의 테스트 가이드라인은 조건문의 분기시키는 조건식 직전값, 조건문의 분기시키는 조건식 직후값. 조건문의 분기시키는 조건식 직전값보다 아주 작은 값, 조건문의 분기시키는 조건식 직후보다 아주 큰 값, 다른 데이터 유형을 수행해 보는 것이다.

반복문의 테스트 가이드라인은 반복문의 초기값, 반복문의 초기값-1의 값, 반복문의 중간값, 반복문의 마지막값, 반복문의 마지막값-1의 값, 다른 데이터 유형이다.

프로그램1의 테스트 데이터는 나이값에 19,20,15,38,스티브잡스, 21.5이다.

나이값에 19를 입력하면 다음 [Figure 4]와 같이 된다.



[Figure 4] Scratch coding- test

나이값이 스티브잡스를 입력하면 티켓가격이 3000원이 되고 나이 변수에 스티브잡스가 입력된다. 그 결과는 다음 [Figure 6]과 같다.



[Figure 5] Scratch coding- age value=19



[Figure 6] Scratchcoding-age value=스티브잡스

나이값에 19.5를 입력하면 나이값 변수에 19.5가 입력되고 출력은 티켓가격은 2400원으로 출력된다.

이와 같이 테스트를 수행한 후에 모든 결과에서 에러가 없다고 판단하면 다양화하기 단계로 넘어간다.

3.3 다양화하기 단계

이 단계에서는 학생들이 1단계에서 개발한 프로그램을 다양하게 변경해 보는 것이다. 다양화하기에는 데이터, 입력출력, 및 알고리즘을 다양하게 적용해 보는 것으로 다양화하기를 할 수 있다.

3.3.1 데이터 다양화하기

나이값에 대해서 할인해 주었는데, 어떤 사람에 대해서 할인해 줄 수 있고, 회원 유형에 따라 할인해 줄 수 있다. 이와 같이 데이터를 다양하게 변경하여 프로그램

을 다양화할 수 있다.

나이값만 아니라 vip 회원에게 할인하는 프로그램으로 변경하여 다양화할 수 있다. [Figure 2]의 스크래치 코드에서 나이가 아니라 회원 유형에 따라 할인하는 프로그램을 만들 때에 다음과 같은 알고리즘을 표현할 수 있다.

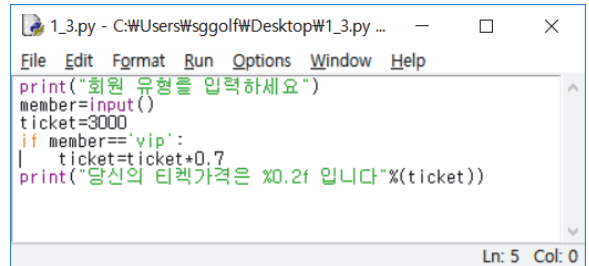
- 문제- 회원유형이 vip이면 50% 할인하는 문제
- 어떤 것을 클릭하면 동작하게 한다. [제어블럭]
- “회원유형을 입력하세요”를 스프라이트가 이야기 하고 답을 입력할 때까지 기다린다.[관찰블럭]
- 회원유형 변수에 입력한 값을 저장된다.[데이터블럭,관찰블럭]
- 티켓변수에 3000원이 저장된다.[변수블럭]
- 회원유형 변수가 “vip”이면 티켓값을 3000×0.7 인 2100원으로 변경한다.[제어 블럭, 연산블럭,데이터블럭][데이터 블럭,연산블럭, 데이터 블럭]
- 티켓값을 10초 동안스프라이트가 이야기 하게 한다.[형태 블럭, 연산블럭,데이터블럭]

위의 알고리즘에 따라 다음 [Figure 7]과 같은 스크래치 코드로 변경할 수 있다.



[Figure 7] Scratch coding-vip

위의 알고리즘에 따라 다음 [Figure 8]과 같은 파이썬 코드로 변경할 수 있다.



[Figure 8] python coding-vip

3.3.2 입력출력 다양화하기

데이터를 입력받을 때에 화면으로 입력받을 수 있거나 파일로 입력 받을 수 있고, 데이터를 출력 받을 때에 화면으로 출력할 수도 있거나 파일로 출력할 수 있게 다양하게 변경해 본다. 스크래치는 보안상의 문제로 온라인 버전에서는 파일로 입출력하는 블록이 없다.

문제나이라는 값을 파일로 입력 받아서 파일로 출력하게 할 수 있다. 이와 같이 처리하는 알고리즘은 다음과 같다.

- 문제- 회원유형이 vip이면 30% 할인하는 문제에서 회원 유형이 텍스트 파일에 저장되어 있을 때

“회원 유형이 있는 파일명을 입력하세요”를 스프라이트가 이야기 하고 답을 입력할 때까지 기다린다.

멤버화일 변수에 입력한 값을 저장된다.
멤버화일 변수의 파일명과 연결할 파일 객체를 만든다.

티켓변수에 3000원이 저장된다.
파일 객체의 값이 “vip”이면 티켓값을 3000×0.7 인 2100원으로 변경한다.
티켓값을 출력하게 한다.

이 알고리즘으로 파이썬 프로그램한 것은 다음 [Figure 9]와 같다.

```

1.3.py - C:\Users\sggolf\Desktop#\...
File Edit Format Run Options Window Help
print("회원 유형인 있는 회원명을 입력하세요")
member=input()
kskim=open(member,"r")
ticket=3000
if kskim.readline()=="vip":
    ticket=ticket+0.7
print("당신의 티켓가격은 %0.2f 입니다"%(ticket))
Ln: 3 Col: 22
    
```

[Figure 9] python coding- file input

```

클릭했을 때
당신의 나이는? 묻고 기다리기
나이 을(을) 대답 로 정하기
티켓 을(을) 3000 로 정하기
만약 나이 > 65 (이)라면
티켓 을(을) 티켓 * 0.7 로 정하기
티켓 가격을 와 티켓 곱하기 을(를) 10 초동안 말하기
    
```

[Figure 10] Scratch coding-condition

3.3.3 알고리즘 다양화하기

알고리즘 다양화하기는 특정 기능을 수행하는 알고리즘을 변경하여 보는 것이다. 또한 새로운 알고리즘을 만들어 보기를 하는 것이다.

문제1에서 변경하여 본다. 나이가 65세보다 크면 가격을 30% 할인해 준다고 변경할 수 있다. 이때에는 조건문의 조건식만 변경하면 된다는 것을 알 수 있게 한다.

이를 알고리즘으로 표현하면 다음과 같다.

- 문제- 나이가 65세 이상이면 30% 할인하는 문제 어떤 것을 클릭하면 동작하게 한다. [제어블럭]
- “나이값을 입력하세요”를 스프라이트가 이야기 하고답을 입력할 때까지 기다린다.[관찰블럭]
- 나이 변수에 입력한 값을 저장된다.[데이터블럭, 관찰블럭]
- 티켓변수에 3000원이 저장된다.[변수블럭]
- 나이 변수가 65보다 크면 티켓값을 3000*0.7인 2100원으로 변경한다.[제어 블럭, 연산블럭,데이터 블럭][데이터 블럭,연산블럭, 데이터 블럭]
- 티켓값을 10초 동안스프라이트가 이야기 하게 한다.[형태 블럭, 연산블럭,데이터블럭]

이 알고리즘 스크래치로 코딩 하면 다음과 [Figure 10]과 같다.

이 알고리즘을 파이썬으로 코딩 하면 다음과 [Figure 11]과 같다.

물론 이와 같은 논리적인 구조가 똑 같은 것은 매우 많은 프로그램을 만들 수 있다.

```

c5_2.py - C:/Users/ssgolf/Desktop/개방형온라인/...
File Edit Format Run Options Window Help
print("나이를 입력하세요")
age=eval(input())
ticket=3000
if age>65:
    ticket=ticket+0.7
print("당신의 티켓가격은 %0.2f 입니다"%(ticket))
Ln: 7 Col: 0
    
```

[Figure 11] Python coding-condition

어떤 값을 입력받아서 특정한 값보다 작으면 ** 한다.

나이가 20보다 적으면 ** 한다.

짝수는 홀수로 만든다.

경차는 주차료를 30% 할인한다.

전기 사용량이 100kw보다 작으면 전기료를 30% 할인해 준다.

아침일찍 오는 손님은 10%할인한다.

3.4 관찰하기 단계

관찰하기 단계에서 학생들이 프로그램을 다양하게 변경하여 보면서 서로 관찰하고 토론하면서 공유하게 한다. 1단계에서는 데이터 변화를 관찰하여야 한다. 본 문제에 대한 데이터를 변화시킨 [Figure2]와 [Figure 7]의

변화를 관찰하고 토론하면서 평가하고 공유하게 한다. 2 단계에서는 다양한 입력의 변화를 관찰하게 하여 평가하고 공유하게 한다. 3단계에서는 알고리즘의 변화를 관찰하여 평가하고 공유하게 한다. 본 문제에 대한 알고리즘을 변화시킨 [Figure2]와 [Figure 10]의 변화를 관찰하고 토론하여 공유하게 한다.

4. IDOL 수업 모형 검증

4.1 검증 방법

본 연구에서 제안한 모형을 검증하기 위하여 초등학교 현장 교사들을 전문가 검증인 델파이 검증을 실시하였다. 초등학교 현장 교사 10명으로 검증하여 그 결과를 반영하였다. 검증 도구는 전용주(2017)의 설문 도구와 강민정(2017)의 설문 도구를 참고로 재구성하였다. 리커트 5점 척도를 활용하였다. 세부 항목에서 교사들의 관점에 평가 항목으로는 타당성(Feasibility-소프트웨어 교육의 수업 모형으로 타당하다), 유용성(Usefulness-소프트웨어 교육을 실행하는데 유용하게 활용될 수 있다), 이해도(Understanding-교사가 쉽게 이해할 수 있다)이다. 학생들의 관점에서 평가 항목은 효과성(Effectiveness-학생들의 창의적인 아이디어를 생성할 수 있다), 자신감(Confidence-학생들이 성취감을 느낄 수 있다), 참여도(Participation-학생들이 적극적으로 참여할 수 있다), 과제집중도(Task concentration-학생들이 문제해결을 위해 집중할 수 있다)이다. 각각의 7개의 항목별로 조사하였다.

4.2 결과 분석

초등교사 10명에 대한 양적 평가는 <Table 1>과 같고, 본 모델에 대한 타당성이 4.80으로 가장 높다는 것을 알 수 있다.

다음은 본 모형에 대한 전문가 설문 결과 각 항목별로 전문가들의 의견은 다음과 같다. 타당성에서는 소프트웨어 교육을 해 보지 않은 교사도 이 모형에 따라 수업에 적용 가능해 보인다. 유용성에서는 초반에 단계

<Table 1> Result

item	average	Standard Deviation
Feasibility	4.80	0.52
Usefulness	4.60	0.52
Understanding	4.60	0.41
Effectiveness	4.30	1.33
Confidence	4.70	0.41
Participation	4.30	0.84
Task concentration	4.60	0.84

적으로 수업을 진행하고 후반부에 관찰을 통해 잘못된 부분에 대한 피드백을 나누는 토론으로 이어지는 수업 흐름이 현재의 교육 방식과 유사하여 수업에 적용하기에 적합해 보인다. 이해도에서는 소프트웨어 교육을 해 보지 않은 교사도 이 모형에 따라 단계적으로 수업에 적용하고 데이터를 다양화하는 방법을 알 수 있을 것 같다.

효과성에서는 예시의 데이터 다양화 방법을 통해서 여러 가지 알고리즘의 다양화 방법을 창의적으로 생각해 낼 수 있을 것 같다. 자신감에서는 쉬운 과제로 성공의 기쁨을 느끼고 알고리즘을 하나씩 추가해 가면서 문제해결력과 성취감을 얻을 수 있을 것 같다. 참여도에서는 단계적으로 교육이 이루어지면서 출력되는 데이터를 예상해보고 결과를 직접 눈으로 보면서 참여 의지를 높일 수 있을 것 같다. 과제집중도에서는 쉬운 과제로 성공의 기쁨을 느끼고 알고리즘을 하나씩 추가해 가면서 문제해결력과 성취감을 얻을 수 있을 것 같다.

5. 결론 및 제언

2015년 개정 교육과정에서 소프트웨어 교육을 해야 한다고 하지만 현장에 적용할 수 있는 모형이 많지 않아 본 연구에서 소프트웨어 교육을 위한 교수 학습 모형인 아이돌 모형을 개발하였다. 본 연구에서 개발한 모형은 구현하기 단계(Implement), 다양하게 변경하기 단계(diversify)와 관찰하기 단계(observe)의 3단계로 구성하는 교수 학습 모형으로 IDOL 모형으로 명명하였다.

제안한 IDOL 모델에 대한 전문가들의 의견을 정리하면 현장 적용 가능성에 대한 타당도가 있고, 수업에 활용할 수 있는 유용성이 있고, 쉽게 이해할 수 있는 모델

이고, 또한 학생들이 창의적인 아이디어를 낼 수 있고, 학생들에게 자신감을 줄 수 있고, 수업에 대한 참여도와 과제집중을 할 수 있는 모형으로 평가하였다.

IDOL 모형을 초등학교 수업 모형에 이용하면 학생이 좋은 아이디어를 만들고, 학생들의 자신감을 주고, 수업에 대한 참여도와 과제집중도를 주기 위한 수업 모델이 될 수 있다.

참고문헌

- [1] CAS, 2013A, Computing in the national curriculum: A guide for primary teachers. Computing At School.
- [2] CAS, 2013B, Computing in the national curriculum: A guide for secondary teachers. Computing At School Computing At School.
- [3] CSTA[2016] CSTA K - 12 COMPUTER SCIENCE STANDARDS REVISED 2016 CSTA STANDARDS TASK FORCE
- [4] CSTA[2017] CSTA K - 12 COMPUTER SCIENCE STANDARDS 2017 CSTA STANDARDS TASK FORCE
- [5] Deborah Seehorn, Stephen Carey, Daniel Moix, Brian Fuschetto, Irene Lee, Dianne O'Grady-Cuniff, Chris Stephenson, Anita Verno(2011). CSTA K-12 Computer Science Standards Revised 2011.
- [6] Deborah Seehorn, Stephen Carey, Daniel Moix, Brian Fuschetto, Irene Lee, Dianne O'Grady-Cuniff, Chris Stephenson, Anita Verno(2016), CSTA K - 12 COMPUTER SCIENCE STANDARDS REVISED 2016 CSTA STANDARDS TASK FORCE
- [7] Kim, Kapsu (2015). A Study on ICT Competences of Korean Students Focus on PISA 2009 and 2012. *Journal of The Korean Association of Information Education*, 19(2), 233-242.
- [8] Kim, Kapsu (2017). An Analysis of Software Curriculum of Korean Elementary Teacher Education School. *Journal of The Korean Association of Information Education*, 21(6), 723-732.
- [9] Kim, Kapsu & Min, Mikyeong(2018) A Study on ICT Competency of Preliminary Teachers Based on PISA 2015 Data. *Journal of The Korean Association of Information Education*, 22(2), 239-249.
- [10] KIM, HYE-SOOK etc(2017) Trend Analysis of ICT Accessibility and Utilization Levels of Korean Students based on OECD PISA Data, *Information policy*. 24(4). 17-43
- [11] ISTE standards(2016) Redefining learning in a technology-driven world A report to support adoption of the ISTE Standards for Students June 2016
- [12] Jeon, YongJu & Kim, TaeYoung(2017), The Analysis of Cognitive and Affective Effects on the CT-CPS Instructional Model for the Software Education Class in Middle School, *Journal of The Korean Association of Computer Education*, 20(4), 47-56
- [13] Lee, Youngoho & Koo, Dukhoi(2015), A Study on the Instructional Design of Software Education Based on Backward Design Model, *Journal of The Korean Association of Information Education*, 19(4), 409-418
- [14] Ministry of Education(2000), Information and Communication Technology Education Guidelines in elementary and secondary schools.
- [15] Ministry of Education(2005), Information and Communication Technology Education Guidelines in elementary and secondary schools.
- [16] Ministry of Education(2015), 2015 Revised National Curriculum 2015-74
- [17] Ministry of Education (2015). The revised national curriculum 2015 for Primary and Secondary Schools, Vol.2015-75, No. 10.
- [18] Sung, Young-Hoon(2015) A Design of Smart-based collaborative learning model for programming education of elementary school, *Journal of The Korea Society of Computer and Information*, 20(4), 147-159

저자소개



김 갑 수

1985.2 서울대학교 계산통계학과
(학사)

1987.2 서울대학교 계산통계학과
전산학전공(석사)

1996.2 서울대학교 계산통계학과
전산학전공(박사)

1987.~1992. 삼성전자 사원-과장

1995.~1998. 서경대학교 전임강
사-조교수

1998.~현재 서울교육대학교 컴퓨
터교육과 조교수-교수

관심분야: 컴퓨터 교육, 소프트웨
어 공학, 정보 영재, 기능성
게임

E-mail: kskim@snue.ac.kr