

하브루타를 활용한 SW 수업 설계 및 적용의 효과성*

김창희**

Effectiveness of Designing and Applying SW Classes Using Havruta

Kim Changhee

〈Abstract〉

This paper is designed to examine the effectiveness of Computational Thinking and programming awareness by designing and applying programming classes using Havruta. In the present society, where the Fourth Industrial Revolution was in full swing, the capacity to be equipped has changed, and the education has been changed accordingly. Programming education is logically capable of thinking and improves comprehensive problem solving skills. This direction of programming education allows us to get ideas for solving problems based on computing thoughts and to create our own creative results. However, because they require the grammar of the programming language and many additional abilities, they are not easy for learners, and individual differences in competencies make learning less immersive and interesting. In this paper, to solve the problems of the uniform programming class, this study is designed and applied to the class applying the Jewish traditional teaching method, Havruta's teaching method, to find out the effect of computing thinking and programming perception.

Key Words : Havruta Teaching, Programming Education, Computational Thinking, Programming Awareness

I. 서론

4차 산업혁명이 본격화된 현재 사회에서는 많은 양의 지식과 정보들이 끊임없이 매일 쏟아진다. 이런 시대에 갖추어야 할 능력들도 많은 변화가 있고, 그에 따른 교육에도 변화가 생겼다. 예측하기 어려운 새롭고 복잡한 문제들의 발생에 따라 기존의 일반적

인 방법을 벗어난 창의적 문제 해결력이 필요하다. 이런 해결력의 일환으로 소프트웨어 교육에서의 프로그래밍 수업이 강조되고 있다[1]. 프로그래밍 교육은 단순히 컴퓨터 언어를 익히는 것을 넘어 논리적으로 사고를 기를 수 있으며 알고리즘에 대한 이해와 데이터 분석능력을 종합적으로 길러 문제 해결 능력을 향상시키는 교육이다[2]. 이러한, 소프트웨어 교육의 방향은 컴퓨팅 사고(Computational Thinking)를 바탕으로 문제 해결을 위한 아이디어를 얻고, 자신만의 결과물을 만드는 과정을 통하여 창의력을 배양하

* 본 연구 내용은 2019년 추계 디지털산업정보학회 학술대회에서 발표한 내용을 토대로 수정·보완한 논문입니다.

** 서울기독대학교 국제경영정보학과 조교수

도록 하고 있다[3, 4].

우리나라에서도 '소프트웨어 중심 사회를 위한 인재양성 추진 계획'에서 제시한 바와 같이, 국가 차원의 정보 교육 과정을 발표하면서 소프트웨어 교육을 강조하고 있고 교육과정도 프로그래밍 중심으로 바뀌고 있다[5]. 하지만 프로그래밍은 프로그래밍 언어의 엄격한 문법 학습 혹은 고도의 추상화 능력을 필요로 하기 때문에 인지적 부담을 증가시켜 학습자가 증도에 포기하도록 하는 결과를 낳는 경우가 많다[6, 7].

또 프로그래밍 교육은 일반적인 컴퓨터실습 수업처럼 획일적 수업의 일변도로 진행되고 학습자의 개인의 역량차가 커서 학습 몰입이 힘들고 그에 따라 학습흥미도가 떨어지게 된다[8].

본 논문에서는 이런 획일적인 프로그래밍 수업에 대한 문제점을 해결 하고자 유대인의 전통적인 교수법인 하브루타 교수법을 적용한 수업을 설계 및 적용하여 컴퓨팅 사고력 측면에서 어떤 효과가 있는지를 알아보려고 한다. 이를 위해서 여러 프로그래밍 언어 중 컴퓨팅 사고력을 효과성을 알아보기 쉽고, 하브루타 교수법을 적용시켜 학습자들간의 대화와 토론을 통해 컴퓨팅 사고력에 대한 변화를 알아보려고 파이썬 프로그래밍 언어를 사용하여 수업 설계 및 적용을 하고 효과성을 알아보려고 한다.

기존의 컴퓨팅 사고력에 관한 연구는 학습요소, 학습도구, 학습 환경들을 고려한 모델개발과 사례분석에 관한 연구들이 2015 SW교육의 의무화 이후에 주를 이루었다. 하브루타 교수법에 대한 기존연구들도 학습흥미도나 학업성취도적인 측면과 동기부여에 관한 연구들이 주를 이루었다. 본 논문에서는 컴퓨팅 사고력 측정도구를 이용하여 하브루타 수업 후에 컴퓨팅 사고력의 구성요소들에 어떤 변화를 가져오는지에 관하여와 하브루타 수업후 프로그래밍에 대한 학생들의 인지도의 변화를 알아보는 연구를 하고자 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구로 SW관점에서의 프로그래밍 교육, 파이썬 프로그래밍 언어, 하브루타 교수법, 컴퓨팅 사고력에 대하여 알아 본다. 3장에서는 연구 방법 및 절차로 연구 대상에 대해 설명하고, 제안하는 수업에 대한 절차와 수업에 대한 수업 지도안들을 설명한다. 제안하는 수업에 대한 효과성 검증을 위해 연구 설계를 하고 컴퓨팅 사고력과 프로그래밍 인식에 대한 정의와 분석을 위한 설문에 대해 알아본다. 4장에서는 하브루타를 활용한 수업을 적용한 결과의 분석과 5장에서는 결론 및 향후과제로 이루어진다.

II. 관련연구

2.1 SW 관점에서의 프로그래밍 교육

세계적으로 SW교육을 강화하고 있는 가운데 우리나라는 CT(Computational Thinking)를 기반으로 문제해결역량을 함양하는 SW교육을 강화하기 위해 소프트웨어교육 운영지침과 2015 개정교육과정 실과/정보과 각론이 발표되었다[9].

2015년 교육부는 2018년부터 SW교육을 의무화 한다고 고시하고 SW교육 인재상 및 교육과정 개편사항을 <표 1> <표 2>와 같이 정의하였다[9].

<표 1> SW교육 인재상

컴퓨팅 사고력을 가진 창의·융합 인재 양성		
초등학교 (체험, 활동)	중학교 (개념 이해)	고등학교 (개발, 융합)
건전한 정보윤리의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해	간단한 알고리즘을 설계하고 프로그램을 개발하여 창의적으로 문제를 해결	효율적인 알고리즘을 설계하고 다양한 분야와 융합하여 문제를 해결

<표 2> SW교육 개편방향

구분	현행	개편안	개편 방향
초등학교 19년~	실과 ICT단원 (12시간)	실과 SW교육 (17시간)	- 문제해결과정, 알고리즘, 프로그래밍 체험 - 저작권 보호 등 정보윤리 의식 함양
중학교 18년~	정보 (68시간 선택)	정보 (34시간 필수)	- 컴퓨팅 사고력 기반 문제 해결 교육 - 알고리즘, 프로그래밍 개 발
고등학교 18년~	정보 (심화 선택과목)	정보 (일반 선택)	- 다양한 분야와 융합한 알 고리즘 설계, 프로그램 개 발 - 기초과목, 실무과목으로 개편하여 NCS 기반 교육과 정 구성

인재상에서는 각 단계에서 필요로 하는 능력을 나타내고 있으며, 개편방향에서는 소프트웨어, 알고리즘과 프로그래밍, 컴퓨팅과 문제해결 3가지 영역으로 나뉘어 3가지 영역에서 필요로 하는 기준을 제시하고 있다. 인재상과 개편방향을 통해 나타내고자 하는 운영지침에서는 컴퓨팅 사고력을 가진 창의·융합인재의 양성, 다시 말해서 단순한 정보통신기술 활용 교육이 아닌 소프트웨어로 새로운 사회의 가치를 창출할 수 있는 컴퓨팅 사고력을 지닌 인재 양성을 목적으로 제시하고 있다[9].

교육부의 의무화 이후 주로 초·중·고등학교의 프로그래밍 교육이 주를 이뤘고, 프로그래밍 언어로는 비교적 학생들이 쉽게 익힐 수 있는 스크래치 언어가 주로 사용되었다.

이애화[10]의 연구에서는 정보교과가 필수과목으로 지정된 2015년 1월부터 2017년 12월 까지 SW교육의 연구동향을 분석 하였는데 그 분석기준은 <표 3>과 같다.

SW교육에 대한 연구주제는 대학을 기준으로 SW교육, 컴퓨팅사고력, 프로그래밍교육의 순으로 나타났고 2018년 이후에는 SW중심 대학 사업이 진행되면서 기존의 전공에서 배우던 프로그래밍 중심의 교

육이 비전공자들에게 확대 되어 그 효과성을 입증하는 연구가 활발하게 이루어졌다[10, 11].

<표 3> SW교육의 연구동향 분석기준

분석 영역	분석기준
연구 대상	유아, 초등학생, 중등학생, 대학생, 교사, 전문가(교수 포함), 일반인 등
연구 방법	개발연구, 문헌연구, 실험연구, 조사연구, 질적연구, 혼합연구
연구 주제	SW교육, 컴퓨팅사고력, 프로그래밍교육, 정보교육, 컴퓨터교육, 코딩교육, 알고리즘교육, 피지컬컴퓨팅, 로봇활용교육, 정보통신윤리, 교수학습방법, 교육과정, 교육 평가 등
SW교육 관련 교과	교과목, 비교과 SW교육(방과 후 교육활동, 동아리 활동, 창의적 체험활동, 교양과목 등)
SW교육 관련 교수학습방법	플립러닝, 스토리텔링, 언플러그드, PBL, 프로젝트학습법, 디자인씽킹, 협동학습 등
SW교육 관련 수업 도구	프로그래밍 도구, 로봇 도구, 3D관련 도구 등

2.2 하브루타(Havruta) 교수법

컴퓨터 실습수업과 같은 개개인의 역량차가 크게 나타나는 수업에서는 학생들의 역량에 따라 학생들의 학습 흥미가 떨어지고 학업 성취도도 개개인의 차이가 클 수 있고, 학습자와의 상호작용이 적기 때문에 학습자를 수동적이고 비참여적인 상태로 만들기 쉽다[12].

하브루타는 유대인의 전통적 학습 방법으로 상호 토론을 통해 자신이 실제 알고 있는 지식을 다시 한번 되짚어 보며, 상대의 지식과 자신의 지식을 비교하며 새로운 지식을 재구성 하는 방식의 교수법이다. 기존 토론식 방법과 다른 점은 1:1로 진행되며 하브루타 학습과정을 거치면서 자신이 교수자와 학습자 역할을 번갈아 가며 진행하며 교수는 주로 수업을 이끌어 주는 역할을 하게 된다[13].

하브루타는 수업의 적용 방식에 따라 <표 4>와 같

이 수업 모형을 구분할 수 있다. 여기서 쉬우르는 수업의 마지막 단계에서 토론과 논쟁의 내용을 교수자가 모든 사람과 나누는 과정으로 질문을 통해 학생들의 사고를 자극해서 해결될 때까지 이끌어 주어야 한다[14].

<표 4> 하브루타 수업 유형

모형	방법
질문 중심	학습내용 읽고 질문 만들기 - 짝토론 - 가장 좋은 질문 선정 - 모둠토론 - 정리발표 - 쉬우르
논쟁 중심	논제 정하기 - 찬성과 반대 입장 정하기 - 짝토론 - 모둠토론 - 정리발표 - 쉬우르
비교 중심	비교할 대상 선정 - 조사하여 질문 만들기 - 짝토론 - 모둠토론 - 쉬우르
친구 가르치기	범위 정해 공부 - 짝 가르치기 - 질문, 대답, 토론 - 바꿔 가르치기 - 질문, 대답, 토론 - 쉬우르
문제 만들기	범위 정해 문제 만들기 - 짝 토론으로 문제 다듬기 - 모둠 토론으로 문제 다듬기 - 발표 - 쉬우르

국내 여러 분야의 교과목에서 하브루타 수업에 대해 다양한 교과목에 특성에 맞춰서 하브루타 교수 방법들이 설계 운영되고 있는데 최근에는 앱 인벤터를 사용한 앱 제작이나 코딩과 관련된 교양 수업에도 하브루타 교수법으로 운영되기도 한다[15, 16].

또, 컴퓨터 실습 수업에 대해 학습몰입과, 학습 흥미도, 학업 성취도 측면에서의 연구도 이루어지고 있다[8].

2.3 파이썬 언어를 이용한 컴퓨팅 사고력

파이썬(Python)은 1989년 네덜란드의 Guido van Rossum이 만든 텍스트 기반의 프로그래밍 언어 중 하나로 파이썬은 다양한 운영체제에서 지원된다. 또 인터프리터식 언어로 컴파일 없이 바로 결과를 확인할 수 있으며 비교적 쉽게 배울 수 있고 그래픽 처리 기능이 단순하기 때문에 초보자가 처음 배우기에 적절하다[17].

파이썬이 프로그래밍 교육용으로서의 적합한 이유는 다음과 같다.

첫째, 파이썬 코드는 읽기가 매우 쉽다. 프로그래밍에 있어 문법의 뜻을 쉽게 이해할 수 있고 TBL(Typing Based Language)중 가장 쉬운 문법을 제공한다.

둘째, 파이썬을 사용하면 다양한 라이브러리를 통해 간결하게 자주 사용하는 어려운 기능을 쉽고 빠르게 구현할 수 있다.

셋째, 배우기가 매우 쉽다[18].

이러한 이유로 미국 내 대학의 경우 프로그래밍을 처음 접하는 학생을 대상으로 프로그래밍 교육을 파이썬으로 운영하는 경우가 69%를 차지한다[19].

컴퓨팅 사고력 CT (Computational Thinking) 는 Wing(2006)에 의해 처음 제시되었고 그 중요성이 부각됨에 따라 많은 컴퓨터 과학자 및 교육자들에 의해 연구 되어졌다[20].

Wing(2006)은 CT가 컴퓨터 과학자의 문제해결 방법을 위한 사고 과정이며 미래에는 모든 사람들에게 필요한 역량이 될 것이라고 주장하였다[20]. 그는 추상화와 자동화가 CT의 하위 요소라고 소개하였다. 그 후 미국의 컴퓨터과학교사협회(CSTA), Google for Education, 영국 등에서 CT의 개념과 하위 요소의 조작적 정의를 세부적으로 정의하였다[21-23].

Google for Education에서는 이러한 CT의 구성요소는 초창기에 분해-패턴인식-추상화-알고리즘-자동화의 5단계로 제시되었고 그 후 더 세부적인 요소들이 정의 되며 CT 교육에 대한 중요성이 부각되기 시작했다[20, 22].

소프트웨어 교육을 통해 학생들의 컴퓨팅 사고력을 향상시킬 수 있는 수업 형태와 수업 모델 관련 연구가 활발히 이루어지고 있으며, 프로젝트 기반 교육, 문제 해결중심 교육, 짝 프로그래밍, 디자인 사고 기법 등 다양한 방법 등이 대표적이다[24, 25].

그러나 이러한 교수학습방법 및 전략은 기존의 보

편적인 교수학습방법들을 SW 교육에 적용한 것이거나 일부 수업전략 형태에 국한되어 있어 SW 교육의 특성을 반영한 교수학습모형으로서는 부족할 수 있다[25]. 이에 한국교육학술정보원은 2015년에 ‘SW 교육 교수학습 모형 개발 연구’를 통해 소프트웨어 교육을 위한 교수학습 모델을 5가지 형태로 제안하고 있다[9].

III. 연구 방법 및 절차

3.1 연구대상

본 연구의 대상은 S대학의 교양 수업을 수강하는 29명을 대상으로 실시하였다. 이 수업은 교양과목의 수업으로 모든 학생이 컴퓨터와 상관없는 전공이며 학년은 다양하였지만 반 이상이 1학년 이었다. 먼저 학생들의 컴퓨터와 관련된 기초적인 지식이 있는지를 조사하기 위하여 현재 학년까지의 컴퓨터 관련 수업 경험을 조사하였다. 초등학교부터 방과 후 수업이나 학원에서의 수업 등을 모두 포함하여 5단계로 나누어서 5점 척도로 점수화 하여 평균을 구하였다. 1단계는 ‘경험이 전혀 없다’, 2단계는 ‘1학기’, 3단계는 ‘2학기’ 4단계는 ‘3학기’, 5단계는 ‘4학기이상’ 수강으로 분류하여 선택하게 하였다. 학원이나 방과 후 수업의 경우 코스 단위를 한학기로 가정하여 선택하게 하였다. 또한, 이 수업을 듣기 전에 컴퓨터 프로그래밍 구현에 대한 자신감에 대해서도 5점 척도로 점수화 하여 조사하였다.

설문 결과 컴퓨터 관련 교육 경험은 <표 5>에서 보는 바와 같이 평균 2.97단계(Sd:1.46)으로 대부분이 수업을 들어본 적은 있는 것으로 나타났으나 프로그래밍 수업으로 한정해서 조사하면 전체 29명 중에 단 2명만이 프로그래밍과 관련된 수업을 들었고 그것도 대학에서의 수업이 아닌 특별활동과 같은 분류의 수

업으로 2명 전부 스크래치에 관한 수업을 들은 것으로 나타났다.

또한, 컴퓨터프로그래밍 구현에 관한 자신감에 대하여서는 평균 1.62(Sd:1.04)으로 매우 낮게 나타났다. 이 결과를 볼 때 본 연구에 참여하는 학생들은 컴퓨터와 관련 없는 전공이며 컴퓨터 프로그래밍을 거의 해본 적이 없는 초보 수준의 학생들임을 알 수 있다.

<표 5 > 사전능력 조사(n=29)

	컴퓨터 수업 경험		프로그래밍 자신감	
	N	%	N	%
1	1	3.4	14	48.3
2	6	20.7	12	41.4
3	16	55.2	3	10.3
4	5	17.2	0	0
5	1	3.4	0	0
M	2.97		1.62	
Sd	1.46		1.04	

3.2 수업 절차

하브루타 수업의 유형중 수업의 진행은 [17]의 수업 유형 중 프로그래밍 수업에 가장 적합한 친구 가르치기 유형으로 진행한다. 수업시간은 도입 단계에서는 전주의 수업내용과 피드백을 하고 그 후에 배우는 수업의 목표와 내용을 간략히 설명한다. 수업에서 간단한 기초 개념 후에 하브루타 교수법을 사용하여 토론을 진행한다.

<표 6> 수업절차

단계	수업 절차
도입	전 주 수업내용 피드백, 수업 목표 및 내용 설명
수업	기초개념 후 하브루타
실습	하브루타와 실습 반복
정리	하브루타 과정을 통한 내용 정리(쉬우르)

<표 7>은 하브루타의 단계별 활동을 나타내는 것

으로서 하브루타 과정에서 1:1 토론을 통해 서로 가르치기-배우면서 질문과 대답, 토론을 통해 내용 파악 - 역할을 바꿔서 반복 하는 과정을 거치고 마지막 쉬우르 과정에서 토론과 논쟁을 통해 합의되고 완성된 내용들을 도출된다면 그것을 공유하는 방식으로 진행을 해나가면 된다.

〈표 7〉 하브루타 단계별 활동

단계	활동
범위정해 공부	해당 단원의 내용에 대해 각자 공부하고 정리 (미리 예습)
짚 가르치기	공부한 내용을 바탕으로 짚 가르치기
질문, 대답, 토론	수업에 대한 내용을 질문, 대답을 통해서 토론하기
바꿔 가르치기	역할을 바꿔서 가르치기
질문, 대답, 토론	수업에 대한 내용을 질문, 대답을 통해서 토론하기 (이해하지 못한 내용 스스로 정리)
쉬우르	교수자가 논쟁과 토론에 관한 것을 모든 사람과 공유

하브루타 과정에서 학습자들은 스스로 모든 것을 진행하는 방향으로 하고 교수자는 이 단계에서 토론 과정의 결과를 도출하지 못하는 경우 중재하거나 아이디어 발견을 위한 촉진자의 역할만을 수행하여야 한다. 학생들은 결과물을 커뮤니티에 올리고 공유한다. 교수자는 하브루타의 결과로 완성된 실습 제출물에 대한 피드백을 다음 단계에서 진행한다.

〈표 8〉은 파이썬을 이용한 프로그래밍 수업의 한 부분으로 하브루타 방식의 수업의 진행을 원활히 하기 위한 수업지도안이다. 도입과 실습 체크 사항을 반복하고 역할을 바꿈으로서 자신의 오류나 문제점들을 스스로 찾아 갈수 있게 하는 방식으로 진행됨으로써 논쟁을 통해서 짚과 합의된 결과를 도출하게 된다.

3.3 연구 설계

본 연구에서 제안하는 수업의 효과성을 분석하기

〈표 8〉 하브루타 수업지도안

단원	변수 이해 및 사용하기. 변수 재활용 하기	주차	1/5
학습 목표	변수 이해, 변수 사용, IDLE 소스편집기 활용, 변수 재활용, 변수 데이터 전달		
단계	짚1	짚2	
도입	학습목표 질문 주의할 점 설명	내용 상기 하며 배우면서 질문	
실습1	변수 이해 및 변수 사용 질문 편집기 활용 질문	짚1 질문에 답변과 제시하는 문제를 직접 실습 하면서 이해 못한 것 질문	
체크 사항	변수의 사용법을 정확히 아는가? IDLE 소스 편집기를 활용할 수 있는가? 이해 못한 내용 정리		
도입	입장 바꿔서 질문		
실습	비교, 논쟁		
실습2	짚2 질문에 답변과 제시하는 문제를 직접 실습 하면서 이해 못한 것 질문	변수 재활용 변수로 데이터 전달 변수명 작성	
체크 사항	변수 재활용 시 주의 사항 변수로 데이터 전달을 할수 있는가 변수명 작성시 주의 사항 이해 못한 내용 정리		
정리	비교 논쟁에 관한 내용과 이해 못한 내용들을 반복하면서 생긴 결론들을 정리		
쉬우르	학생들이 자유롭게 생각한 것을 이야기 하도록 이끌고 내용을 공유 및 피드백		

위하여 컴퓨팅 사고력의 향상과 프로그래밍에 대한 인식을 연구모델로 결정하고 수업 전에 비해 수업 후에 역량 및 인식 향상에 효과가 있는지를 확인하기 위해 아래와 같은 연구문제를 설계하였다.

〈표 9〉 연구 설계

O ₁	X ₁	O ₂
----------------	----------------	----------------

X1 : 하브루타를 활용한 SW수업
O1 : 사전검사(컴퓨팅 사고력, 프로그래밍 인식)
O2 : 사후검사(컴퓨팅 사고력, 프로그래밍 인식)

가설 1: 하브루타를 활용한 SW수업은 컴퓨팅 사고력 향상에 유의미한 영향을 미칠 것이다.

가설 2: 하브루타를 활용한 SW수업은 프로그래밍 인식의 변화에 유의미한 영향을 미칠 것이다.

수업을 듣기 전 사전 평가에 비해 수업 후의 평가의 평균이 높고, 그 차이가 통계적으로 유의미하다면 본 연구에서 제안하는 수업이 효과성이 있다고 볼 수 있다.

3.3.1 컴퓨팅 사고력

컴퓨팅 사고력 검사를 위한 검사지는 한국교육학술정보원에서 제공하는 컴퓨팅 사고력 검사지를 본 연구에 맞게 구성하여 검사도구로 사용하였다[26].

사전과 사후로 나뉘어 검사를 하였고 각 SW 역량에 대한 컴퓨팅 사고력의 구성요소는 아래 <표 10>과 같다.

<표 10> 컴퓨팅 사고력 측정 도구

SW 역량	분석 능력	설계 능력	구현 능력	추론 능력
컴퓨팅 사고력 구성요소	자료 수집	추상화	자동화	시물레이션
	자료 분석			
	자료 표현	알고리즘		병렬화
	문제 분해			

위의 표를 바탕으로 컴퓨팅 사고력 구성요소 중 규칙, 추상화, 알고리즘, 자동화, 시물레이션 등 5가지의 컴퓨팅 사고력 구성요소에 대해 사전 사후 검사를 통해 통계적인 차이를 확인하였다.

3.3.2 프로그래밍 인식

프로그래밍 인식에 관한 검사지는 [2]를 참고하여 크게 3가지 영역으로 구분하여 사전과 사후 각각 9개

의 문항으로 구성하였고 3가지 영역은 ‘프로그래밍에 대한 기초정보 및 흥미 변화’, 프로그래밍의 가치와 중요성에 대한 인식 변화’, 프로그래밍과 미래사회 직업에 대한 인식변화’로 나뉘어 사용하였다.

IV. 연구결과 및 논의

본 연구는 하브루타를 활용한 SW수업을 설계하고 그 수업을 진행함에 따라서 컴퓨팅 사고력과 프로그래밍 인식에 어떤 영향을 미치는지를 알아보려고 하였다.

각 도구는 컴퓨팅 사고력과 프로그래밍 인식에 대해 사전검사, 사후검사를 통해 통계적인 차이가 있는지를 분석하였다. 수업의 효과를 알아보기 위하여 SPSS 21을 이용하여 t-검증을 수행하였다. 각 도구별 사전 검사에 독립표본 t-검증을 실시하였고, 사후 검사 결과가 통계적인 유의미한 차이를 알아내어 위해 앞의 가설이 증명되는지를 밝히고자 하였다.

4.1 컴퓨팅 사고력

컴퓨팅 사고력에 관한 사전검사와 하브루타를 활용한 수업을 적용 후에 학생들의 컴퓨팅 사고력에 대해 통계적인 차이가 있는지를 알아보기 위해 사전·사후 독립표본 t-검증을 실시한 결과는 <표 11>과 같다.

컴퓨팅 사고력의 규칙은($t=3.249, P=.001$) 유의 확률이 .05보다 작으므로 통계적으로 유의미한 변화가 있는 것으로 나타났다. 또 추상화($t= 2.789, P=.006$), 알고리즘($t= 2.135, P=.035$), 시물레이션($t= 2.581, P=.017$)에서도 통계적으로 유의미한 결과를 나타냈다. 다만 자동화 ($t= 1.128, P=.114$)는 유의미한 통계적 차이는 나타나지 않았는데, 학생들이 자동화 부분의 반복문과 관련된 부분들을 어려워하는 것으로 나

<표 11> 컴퓨팅 사고력 결과(N=29)

구성요소		M	Sd	t	P
규칙	사전	67.24	6.92	3.249	.001
	사후	78.53	11.06		
추상화	사전	62.51	5.85	2.789	.006
	사후	73.21	10.31		
알고리즘	사전	57.82	7.24	2.135	0.35
	사후	64.54	6.98		
자동화	사전	55.65	6.29	1.128	.114
	사후	60.07	5.16		
시뮬레이션	사전	63.27	6.18	2.581	.017
	사후	71.06	9.27		

타났다. 프로그래밍을 처음 접해보는 상황에서 알고리즘과 자동화 부분이 상대적으로 컴퓨팅 사고력 부분 향상에 낮은 영향을 미치는 것으로 나타났다. 하지만 대부분의 컴퓨팅 사고력 구성요소에서 통계적으로 유의미한 결과를 나타냈기 때문에 ‘하브루타를 활용한 SW수업은 컴퓨팅 사고력 향상에 유의미한 영향을 미칠 것이다’라는 [가설 1]이 성립됨을 보여준다.

4.2 프로그래밍 인식

프로그래밍 인식에 관한 사전검사와 하브루타를 활용한 수업을 적용 후에 학생들의 프로그래밍 인식에 대해 통계적인 차이가 있는지를 알아보기 위해 독립표본 t검정을 이용하여 실시하였다. 영역별 통계결과는 <표 12>에 나타났다. 세부 항목을 보면 1영역 ‘프로그래밍에 대한 기초정보 및 흥미 변화’에서 유의 확률이 .05 보다 작으므로 통계적으로 유의미하게 나타났다. 2영역 ‘프로그래밍의 가치와 중요성에 대한 인식변화’ 항목에서도 유의 확률 .05보다 작으므로 통계적으로 유의미하게 나타났다. 3영역 ‘프로그래밍과 미래사회 직업에 대한 인식변화’은 향상은 되었지만 유의미한 차이는 나타나지 않았는데, 이는 프로그래

밍과 미래직업 인식에는 아직 학생들에게 크게 작용하지 않음을 증명하고 있다. 프로그래밍이 흥미와 중요성은 인정하지만 아직 프로그래밍으로 인한 직업에 대한 인식은 좀 더 시간과 교육이 더 필요 할 것으로 보인다. 하지만 대부분의 항목에서 ‘하브루타를 활용한 SW수업은 프로그래밍 인식의 변화에 유의미한 영향을 미칠 것이다.’ 라는 [가설 2]가 성립됨을 보여준다.

<표 9> 프로그래밍 인식 결과(N=29)

구성요소		M	Sd	t	P
프로그래밍에 대한 기초정보 및 흥미 변화	사전	3.38	1.042	2.394	.014
	사후	4.09	.751		
프로그래밍의 가치와 중요성에 대한 인식변화	사전	3.87	.641	3.817	.001
	사후	4.12	.574		
프로그래밍과 미래사회 직업에 대한 인식변화	사전	3.57	.982	-0.284	.167
	사후	3.81	.871		

4.3 하브루타 수업에 대한 학생의견

사후 조사들이 끝난 후 간단한 설문 문항으로 하브루타를 활용한 SW 수업에 대한 반응을 살펴보았다. 설문 문항은 아래와 같다.

1. 이 수업으로 프로그래밍이 무엇인지 알게 되었다.
2. 이 수업으로 프로그래밍이 우리생활에서 중요하다고 생각되었다.
3. 이 수업으로 프로그래밍과 관련된 직업을 갖고 싶어졌다.
4. 이 수업으로 미래에는 프로그래밍 능력이 필요하다는 것을 느꼈다.
5. 이 수업을 통해 프로그래밍에 대한 흥미와 자신감이 생겼다.

설문에 대한 응답은 아래 <그림 1>과 같다. 각 질문에 대한 긍정적 대답인 “매우 그렇다”와 “그렇다”의 평균은 88.2%로 하브루타 활용한 SW 진행한 학생들은 그 효과에 대해 긍정적으로 생각 하는 것으로 나타났다.



<그림 1> 하브루타 수업에 대한 학생의견

그 외 주관식 응답은 다음과 같다.

<긍정적 의견>

다른 수업과 다르게 질문을 학생스스로가 만들어서 이야기 하는 방식이 흥미롭다고 생각합니다.

수업을 하면서 평소 잘 알지 못했던 친구들과 친해 질수 있었습니다.

프로그래밍이라는 것이 생소하였는데 흥미가 생겼습니다.

수업시간에 집중력이 높아진 것 같습니다.

수업시간에 내 의견을 적극적으로 표현할 수 있었습니다.

토론을 통해서 프로그래밍에 관한 내용들을 수정 할 수 있었습니다.

다른 수업시간에는 수동적인 자세로 수업을 했었는데 이 수업은 능동적인 자세를 가지게 되었던 것 같습니다.

<부정적 의견>

프로그래밍에 대한 기본지식이 없어서 토론하기가

쉽지 않았습니다.

친구와의 평소 관계에 따라 수업시간의 진행이 큰 영향을 미치는 것 같습니다.

모르는 것이 나왔을 때 어떻게 해야 할지 잘 모르겠습니다.

평소 사이가 좋지 않았던 친구와는 수업에 대한 진행이 어려웠습니다.

이 수업에서 교수님의 역할은 무엇인지 잘 모르겠습니다.

주관식 응답에서 긍정적 의견과 부정적 의견들이 나왔는데 긍정적 의견은 대부분 흥미로움과 수업에 임하는 자세들이 좋아졌다는 의견이고, 부정적 의견은 하브루타의 개념이 아직 명확하게 인지하지 못하는 상황에서 오는 의견으로 판단된다.

V. 결론 및 향후과제

4차산업혁명이 본격화된 현재 사회에서는 매일 쏟아지는 많은 양의 지식과 정보들에 따른 교육에도 큰 변화가 생겼다. 예측하기 어렵고 복잡한 문제들의 발생에 따라 기존의 방법으로는 해결하지 못하기 때문에 창의적인 문제 해결방법이 필요하다. 소프트웨어 교육에서 프로그래밍 수업이 강조되는 이유이다. 하지만 프로그래밍 수업도 일반적인 컴퓨터 실습수업처럼 개인적인 능력과 역량의 차가 큰 학습자들을 대상으로 수업을 진행하면 창의적인 문제 해결방법이 힘들고 컴퓨팅 사고력의 증진에도 도움이 될 수 없다. 하브루타 교수법은 특히 학습자들의 수업에 대한 몰입과 흥미도 측면에서 좋은 결과를 가져오는 학습 방법으로 다양한 분야에서 사용되었다. 학습자가 서로 상호 간의 토론을 통해 상대의 지식과 자신의 지식을 비교하며 새로운 지식을 재구성하는 하브루타 방식을 SW 수업에 도입하는 것은 컴퓨팅 사고력 증진과

프로그래밍 인식에 영향을 줄수 있다. 본 연구에서는 파이썬 프로그래밍 언어를 사용하는 프로그래밍 수업에 하브루타 방식의 학습법을 설계하여 학생들이 토론을 거쳐 교수자와 학습자의 역할을 번갈아 가며 하게 됨으로써 그 과정 속에서 컴퓨팅 사고력 향상시키고, 프로그래밍에 대한 인식이 통계적으로 유의미함을 비교하였다.

제안하는 수업을 적용한 연구의 결과 첫째, 하브루타를 활용한 SW수업은 컴퓨팅 사고력 향상에 유의미한 영향을 미쳤다. 둘째, 하브루타를 활용한 SW수업은 프로그래밍 인식의 변화에 유의미한 영향을 미쳤다. 셋째, 하브루타 방식의 수업으로 인하여 평소 어렵게 느껴졌던 프로그래밍에 대한 거리감을 조금이나마 줄일 수 있었다. 기존의 학습법들에서 플립러닝, PBL같은 학습법들이 주로 학업 성취도 면에서 연구를 하였고 하브루타 방식의 학습법에 대한 프로그래밍 연구들이 이루어지고 있는 시점에서 본 논문은 대학생들이 가장 많이 사용하는 파이썬 프로그래밍을 이용하여 컴퓨팅 사고력에 향상에 영향을 미치는 것을 분석하였다.

수업에 대한 효과성은 통제 집단과 실험집단을 구분하여 논문에서 제안하는 수업을 각각 적용시켜서 수업 전과 후에 대한 결과를 나타내는 것이 '학업성취도'와 같은 측면에서는 논문의 효과성을 보다 효율적으로 나타낼 수 있다. 하지만 이와 같은 실험은 한 학기에 같은 집단을 나눠서 실험해야 하는 등, 제약이 있어 컴퓨팅 사고력과 프로그래밍 인식에서의 측정으로 대신 하였다. 이는 수업을 적용한 집단의 학생들이 수업을 하기전과 수업을 한 후에 나타나는 변화로 하브루타 수업의 효과성을 나타낼 수 있지만, 집단을 실험과 통제집단으로 나눠서 효과를 측정할 것에 비해서는 연구의 한계로 작용할 수 있다.

향후 과제로는 하브루타의 수업에 대한 보다 전문적인 수업 지도안의 개발이 필요하다. 전통적으로 주입식 교육을 받아온 우리교육의 현실에서 하브루타

방식의 교육은 많은 학생들이 처음에 낯설어 하고 적응에 힘들어 한다. 1:1 토론방식의 수업이다 보니 다른 외부요인에 의해서 수업의 효과성이 바뀌기도 한다. 초·중·고등학교의 SW의무 교육의 일환으로 많은 연구들이 이루어지고 있다. 대학들도 SW중심 대학 사업의 일환으로 SW교육과 관련되어 프로그래밍과 다양한 방법의 연구들이 이루어지고 있다. 각 교수법마다 프로그래밍 수업과의 연계성을 찾아보고 다양한 변인들을 추가하여 창의성과 문제해결능력을 갖출 수 있는 SW 수업에 관한 연구가 필요할 것으로 생각된다.

참고문헌

- [1] 김창희·류정혜, “하브루타 기반 프로그래밍 교육에 대한 효과,” 디지털산업정보학회 학술대회발표논문집, 2019 공동학술대회
- [2] 김성경, “중학교 프로그래밍 수업에서의 하브루타 교수법을 활용한 수업설계 및 적용,” 한국외국어대학교 교육대학원, 석사학위논문, 2018.
- [3] 이희원·민혜리·이경우, “공과대학 교양교육 개선 방안 탐구:서울대학교 사례를 중심으로,” 공학교육연구, 제11권, 제3호, 2008, pp.1-9.
- [4] 정해원, “IT 유창성과 컴퓨팅적 사고 개념을 이용한 대학 정보교양 교육에 관한 실증적 연구,” 한국컴퓨터정보학회논문지, 제19권, 제2호, 2014, pp.263-274.
- [5] 공진호·손종걸, “소프트웨어중심사회를 위한 인재양성 추진계획,” 과학기술정보통신부 SW정책과, <http://www.msit.go.kr/web/msipContents/contentsView.do?cateId=mssw315&artId=1270998>
- [6] 배학진·이은경·이영준, “문제 중심 학습을 적용한 스크래치 프로그래밍 교수 학습 모형,” 컴퓨터교육학회논문지, 제12권, 제3호, 2009, pp.11-22.

- [7] 홍성권 · 최정원 · 이영준, “초등학생의 게임 프로그래밍 경험이 자기효능감에 미치는 영향,” *교육교육*, 제30권, 제3호, 2014, pp.197-215.
- [8] 김창희, “컴퓨터 실습수업에서 하브루타 교수법 효과에 따른 연구,” *디지털산업정보학회 논문지*, 제14권, 제4호, 2018, pp.177~187.
- [9] 김진숙 외, “SW교육 교수학습 모델 개발 연구” *한국교육학술정보원*, 2015년 교육정책네트워크 교육현장지원 연구, 수탁연구 CR2015-35
- [10] 이에화, “국내 소프트웨어교육 연구동향 분석,” *교육정보미디어연구*, 제24권, 제2호, 2018, pp.277-301.
- [11] 김민자 · 김현철, “컴퓨팅 사고력 관점에서 본 컴퓨터 비전공자 대상 교양 컴퓨팅 수업의 효과성 분석 연구,” *컴퓨터교육학회 논문지*, 제21권, 제1호, 2018, pp11-21.
- [12] 전정희, “또래교수법을 실시한 컴퓨터 실습수업에서 동료평가가 학업성취도와 자기효능감에 미치는 영향”, *고려대학교 교육대학원 석사학위*, 2011.
- [13] Kent, O., “A theory of Havruta learning”, *journal of jewish education*, Vol,76, No3, 2010, pp.215~245.
- [14] 전성수, *하브루타로 교육하라 : 질문하고 토론하는 하브루타 교육의 기적*, 위즈덤하우스, 2012.
- [15] 김정숙 · 이태욱, “하브루타를 접목한 앱 제작으로 컴퓨팅 사고력 신장”, *한국컴퓨터정보학회 동계 학술대회 논문집* 제26권, 제1호, 2018, pp.223-226.
- [16] 이에리, “플립러닝과 하브루타 학습법에 기반한 학습자 중심의 코딩 수업 설계 및 적용”, *디지털산업정보학회 논문지*, 제14권, 제2호, 2018, pp.69~78.
- [17] Python Software Foundation. 2017. Python about. <https://www.python.org/about/>
- [18] 이대현, “파이썬 언어를 활용한 게임 프로그래밍 교육,” *한국게임학회지*, 제11권, 제1호, 2014, pp.8-16.
- [19] P. Guo. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities, *BLOG@CACM*, 2014, <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>.
- [20] M. Wing, “Computational thinking and thinking about computing. Philosophical transactions of the Royal Society,” *A*, 366, 2008, pp.3717-3725.
- [21] Department for Education. (2013). The national curriculum in England: Framework document. Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210969/NC_framework_document_-_FINAL.pdf.
- [22] Google for education(2015). CT Lesson Plan. <https://www.google.com/edu/resources/programs/>
- [23] International Society for Technology in Education & Computer Science Teachers Association. (2011). CSTA K-12 Computer Science Standards Revised 2011. Retrieved from http://csta.acm.org/Curriculum/sub/urrFiles/CSTA_K-12_CSS.pdf.
- [24] 김종우, “2009 교육과정에서 컴퓨터교육의 적용,” *교육과학연구*, 제12권, 제1호, 2010, pp.41-55.
- [25] 전수진, “SW 교육에서의 컴퓨팅 사고력 신장을 위한 개발 중심 모형의 설계 및 효과,” *정보교육학회논문지*, 제21권, 제6호, 2017, pp.619-627.
- [26] 안성훈 외, “2016년도 SW교육 연구학교 효과성 분석 연구,” *한국교육학술정보원*, 연구보고 KR2016-4, <http://lib.keris.or.kr/search/detail/CATLAB000000012086>.

■ 저자소개 ■



김 창 희
Kim Changhee

2008년 3월-현재
서울기독대학교 국제경영정보학과
교수
2007년 2월 명지대학교 컴퓨터공학과
(공학박사)
2000년 2월 명지대학교 컴퓨터공학과
(공학석사)
1998년 2월 명지대학교 컴퓨터공학과(공학사)
관심분야 : 컴퓨터 교육, Social Computing,
e-learning
E-mail : area88@scu.ac.kr

논문접수일 : 2019년 11월 28일
수 정 일 : 2019년 12월 12일
게재확정일 : 2019년 12월 12일