

# Storytelling 기반 SW교육이 Computational Thinking에 미치는 영향

박정호

가수초등학교

## 요 약

개인의 아이디어 구현이나 문제해결에 SW를 중심으로 활용하는 SW중심사회로의 진입으로 SW교육이 강조되고 있다. 일반적으로 스토리텔링은 학습자의 흥미를 끌며 맥락적 환경을 제공하는 것으로 알려져 있다. 본 연구는 이솝우화를 활용한 SW교육프로그램이 Computational Thinking에 미치는 효과를 알아보기 위해 실험연구를 실시하였다. 총 6주 동안 토요일 SW교육 캠프에 참여한 학생 중 실험집단은 이솝우화를 통해 스크래치 프로그래밍의 기초개념을 배운 후 이야기, 게임, 예술, 시뮬레이션 등의 다양한 SW표현활동을 경험하였다. 프로그램 적용 후 실험 및 비교집단을 대상으로 SW개념, SW구현 그리고 SW교육에 대한 태도를 비교 검증하였다. 연구결과 실험집단에 참여한 학생들의 SW개념과 SW구현 결과가 더 높게 나타났다. 그리고 SW교육에 대한 태도에서도 더 긍정적인 결과를 확인하였다.

키워드 : 스토리텔링, 소프트웨어교육, 이솝우화, Computational Thinking, 스크래치

## Effects of Storytelling Based Software Education on Computational Thinking

Jungho Park

Ga-su Elementary School

## ABSTRACT

Since the entering SW-centered society in which people make the best use of SW in order to express their own personal ideas or to solve problems, the significance of SW education has been emphasized. It is generally known that storytelling provides motivational and contextual environments. This study conducted an experimental research to find out how Aesop's fables-applied SW educational program would influence computational thinking. Of these students who had participated in an SW educational camp on every Saturday for a total of 6 weeks, the experimental group learned on basic concepts of Scratch programming as working on Aesop's fables and they were led to diverse SW activities such as stories, games, simulations arts, and others. After that, the study carried out a comparison research in relation to the groups' SW concepts, SW implementation and attitudes toward SW education. The result shows that the experimental group came up with higher levels of SW concepts, implementation and positive attitude toward SW education.

Keywords : Storytelling, Software Education, Aesop's Fables, Computational Thinking, Scratch

논문투고 : 2015-01-29

논문심사 : 2015-01-29

심사완료 : 2015-03-18

## 1. 연구의 필요성 및 목적

현 사회는 IT를 기반으로 하는 지식정보화 사회를 뛰어넘어 소프트웨어(이하 SW)가 경쟁력을 결정하는 'SW중심사회'로 진입하고 있으며 세계 주요국들은 이러한 변화에 적극 대응하기 위해 SW에 기반한 다양한 국가 전략을 추진하고 있다. 'SW중심사회'란 SW를 통해 개인, 기업, 정부 전반에 걸친 삶의 질이 향상되고 아이디어와 상상력을 구현하거나 다양한 문제점 해결에 SW를 중심으로 활용하는 사회를 의미한다[18].

한편 영국은 기존의 ICT교과를 알고리즘 이해, 프로그램 작성 및 디버깅, 프로그램 결과 예상을 위한 논리적 사고, 데이터구조 등의 컴퓨터과학에 근거한 컴퓨팅 교과로 변경하였으며 K-12학생들이 필수적으로 SW교육을 이수하도록 하는 명실상부한 SW교육의 선도국가로써의 위상을 다지고 있다[9].

SW교육은 전통적인 수학, 과학, 공학, 사회 등의 교과에서 다루지 않던 새로운 유형의 사고방식인 Computational Thinking(이하 CT)을 기를 수 있다. Kazakoff(2013)에 따르면 애니메이션, 그래픽 모델, 게임 그리고 나이에 적합한 프로그래밍 학습이 추상화, 자동화, 분석, 문제분해, 모듈화 그리고 반복적 설계와 같은 핵심적 CT개념들을 배우고 적용하는 기회를 제공한다고 하였다[11].

1980년대의 LOGO 프로그래밍 이후 SW교육 도구는 양적, 질적으로 발전하고 있다. 예로 스크래치, 러플(Rur-Ple), Tynker, E-toys 등과 같은 다양한 종류의 EPL, 알고리즘 사고 및 프로그래밍 개념 이해를 돕는 Light-Bot, FixTheFactory와 같은 코딩게임, 그리고 다양한 창작활동을 지원하는 피코보드, 위두, 마키마키, 로봇 등의 구체적 조작도구의 등장은 SW교육환경을 풍요롭게 하고 있다.

이중 스크래치는 스토리, 애니메이션, 게임, 예술 등의 학습자 주도의 창작활동을 가능하게 하는 대중적 SW교육 도구로 알려져 있다. 하지만 좋은 기술이나 도구를 사용하는 것만이 항상 긍정적 교육결과를 가져다 주는 것은 아니다. 따라서 많은 연구자들은 SW교육을 발전시키기 위한 새로운 형태의 교육방법, 교재개발, 교육환경 개선 등에 관한 꾸준한 연구를 수행하고 있다. 즉, '좋은 도구를 갖고 어떻게 교육할 것인가?'에 대한

답인 SW교육에 학습자를 능동적으로 참여시킬 수 있는 교수학습방법의 구안이 중요한 과제라 할 수 있겠다.

더욱이 초등학교는 SW교육을 처음 접하는 시기이고 향후 직업진로 선택에 영향을 미칠 수 있으므로 SW교육 활동 자체에 흥미를 갖게 하는 것은 매우 중요하다고 볼 수 있다.

이러한 측면에서 학습자의 흥미를 끌며 맥락을 제공하는 스토리텔링은 SW교육에서 교수학습방법의 하나가 될 수 있다. 특히 초등학생에게 친숙한 이솝우화의 활용은 긍정적 결과를 기대할 수 있을 것이다. 이에 본 논문은 이솝우화를 활용한 SW교육이 CT에 미치는 효과를 알아보기 위해 다음의 연구를 실시하였다.

첫째, 문헌연구를 통해 CT, SW교육, 이솝우화를 고찰하였다.

둘째, SW교육 프로그램을 개발한 후 실험연구를 실시하였다.

셋째, SW개념 평가, SW구현능력 평가 그리고 SW교육에 대한 태도에 관한 효과성 검증을 실시하였다.

## 2. 이론적 배경

### 2.1 Computational Thinking

Wing(2006)이 21세기를 살아가야 하는 모든 사람이 갖추어야 할 필수 사고로 CT를 발표한 후 정보교육에서 CT의 역할과 의미가 큰 주목을 받아오고 있다[21].

K-12교육에서 교과를 초월하여 자동화, 전이 그리고 적용할 수 있는 문제해결방법으로 CT에 대한 역할이 부여되고[3], 모든 학문에 필수적이라고 제안하였다[5]. 하지만 CT가 구체적으로 무엇을 포함하고 있는 것에 대해선 아직 학자들 간의 일치된 합의는 도달하지 못한 상황이다[1][2]. 이것은 CT의 개념과 구성요소에 대한 학자나 관련 단체들의 차별화된 정의에서 나타나고 있다.

CT의 기본 개념은 추상화, 알고리즘, 평가, 문제분해, 일반화라 정의되기도 하였으며[8], Cuny 외(2010)는 정보기기에 의해 솔루션을 효과적으로 실행될 수 있는 형태로 표현하기 위해 문제를 공식화하고 솔루션을 만드는 것을 CT로 정의하였다[7].

한편 CSTA(computer science teachers association)

에서는 CT의 특징은 컴퓨터를 사용할 수 있도록 문제를 공식화하기, 데이터의 논리적인 조직과 분석, 추상화를 통한 데이터 표현, 알고리즘 활용 자동화된 솔루션 개발, 최적화된 솔루션 구현, 솔루션의 일반화로 정의하였다[6].

여러 학자들의 견해에 비추어 볼 때 CT는 정보교육을 넘어 모든 교과에 적용할 수 있는 새로운 형태의 사고체계로써 추상화, 알고리즘, 문제분해, 논리, 알고리즘, 자동화, 일반화 등의 하위 사고능력을 포함하여 문제를 해결하는 능력이라 할 수 있고 이러한 사고들은 순서, 반복, 조건, 연산, 함수 등의 프로그래밍 개념과 도구를 통해 구현될 수 있다. 그리고 K-12교육과정에서 CT를 성공적으로 진입시키기 위해서 CT를 어떻게 측정할 것인가, 즉 학생들이 배운 것을 교사가 평가하는 타당성 확보는 선결되어야 할 과제이다.

CT평가와 관련한 대부분의 연구는 Werner(2012)의 동화 평가(fairy assessment)처럼 추상화, 조건 논리, 알고리즘 사고 그리고 문제해결을 위한 다른 CT의 개념 이해와 활용을 평가하기 위해 학생이 창작하거나 미리 계획된 프로그래밍 결과물을 활용하였다[19]. 예로 Malyn-Smith 외(2010)는 조건과 파라미터 학습내용 평가에서 짧은 이야기 형식의 문제를 제공한 후 오류를 찾고 처치하도록 하였으며 과제 수행결과에 따라 CT의 수준을 상, 중, 하로 분류하였다[15]. 한편 Brennan & Resnick(2012)는 스크래치에서 CT 평가방법으로 프로젝트 포트폴리오 분석, 산출물 기반 인터뷰, 설계 시나리오 분석들을 제시하였다[4]. 그리고 Fields 외(2012)는 학습자에게 오류가 있는 프로젝트를 디버깅하게 함으로써 엔지니어링(engineering)과 프로그래밍 능력을 평가하였다[10].

초등학생을 대상으로 CT를 측정하기 위해 개발된 공식적인 평가도구는 찾기가 쉽지 않다. 다만 다른 유형의 사고력 측정 검사도구와 마찬가지로 복잡한 CT 측정을 위해 SW개념 평가와 SW구현능력 평가가 혼용되어 사용되어지고 있는 것으로 알려져 있다.

## 2.2 이솝우화 활용 스토리텔링

스토리텔링은 이야기를 만들고 타인에게 서사적으로 표현, 전달하는 행위를 지칭하는 것이다.

SW교육에서도 스토리텔링과 관련한 연구가 수행되고 있다. 예로 디지털 스토리텔링이 제공하는 프로그래밍 활동에서도 학습의 ‘즐거움’을 증가시키고[17], 학습동기와 성취도를 향상시킨다는 연구들이 수행되었다[13]. 또한 스토리텔링을 기반으로 한 로봇 교육과정을 설계하고 이를 지원할 수 있는 시스템 개발에 관한 연구도 수행되었다[14]. 하지만 선행연구들은 스토리텔링을 학습흥미를 끄는 매개체로 활용하였지만 직접적으로 SW개념, 즉 학습 내용과 관련이 있는 것은 아니었다.

단순히 이야기를 들려주고 SW로 표현하는 것을 넘어 학생들이 배운 CT개념과 직접적으로 관련 있는 이야기를 제시함으로써 CT를 보다 즐겁고 맥락적으로 이해하게 한다면 학습자의 능동적 참여를 이끌어 낼 수 있을 것으로 보인다.

선행연구 분석 결과 이솝우화를 이용하여 SW교육을 실시하거나 이야기의 구조를 CT의 개념과 관련지은 연구는 없었다. 이솝우화는 동물이 약 70% 정도 등장하고 의인화된 표현이 많아 초등학생에게 친숙하다. 또한 교훈을 주는 짤막한 이야기로 구성되어 있어 학습자들은 부담을 느끼지 않고 자연스럽게 몰입할 수 있다. 그리고 친숙한 이야기 속의 한 장면을 SW로 표현해보는 다양한 활동은 학습동기를 유발시킬 수 있다.

## 2.3 SW교육

프로그래밍교육 또는 코딩교육과 동일한 의미로 받아들여지는 SW교육의 역사는 1980년대 Papert LOGO 프로그래밍 시대로 거슬러 올라간다.

초기에 SW교육이 교육현장에 정착하지 못한 이유는 응용프로그램의 대중화로 프로그래밍 교육의 필요성이 쇠퇴한 외부 요인과 함께 몇 가지 내부적인 한계를 갖고 있었다. 첫째, 프로그래밍을 다양한 창작표현의 도구가 아닌 프로그래밍 그 자체로 인식하였다. 예로 어린이들의 흥미나 경험과는 거리가 먼 선, 도형 그리기와 같은 활동이 대부분이었다. 둘째, 아동들의 프로그래밍 학습을 돕는 환경이 미약하였다. 예를 들어, 오류가 발견되었을 때 조언이나 성공하였을 때 확장된 탐구과제를 제공할 수 있는 지원체제가 부족한 환경이었다.

교육용프로그래밍 언어는 누구나 쉽게 배우고, 질 높은 기능을 구현하며 그리고 다양한 영역의 표현활동을

지원해야 한다. Kelleher & Pausch(2005)에 따르면 논리적이고 추상적 사고를 동반하는 정보처리과정에 익숙하지 않은 초등학생에게 프로그래밍의 개념, 원리 및 기능, 실행 방식과 구문 등을 동시에 학습하는 것은 심리적 부담을 안겨줄 수 있다고 하였다[12].

이와 같은 측면에서 스크래치는 초등학생에게 적합한 SW교육 도구로 볼 수 있다. 그리고 스크래치는 확장기능을 지원하고 있어 실세계와 연계된 창의적 SW 문제해결활동을 가능하게 한다.

### 3. 연구방법

#### 3.1 연구절차

본 연구는 2014년 11월 8일부터 12월 13일까지 6주 동안 주당 3시간씩 토요일 스크래치 캠프를 통해 수행되었다. 연구대상은 경기도 D초등학교 5, 6학년 248명을 모집단으로 연구대상은 식별번호를 부여받고 무선으로 실험 및 비교집단에 20명씩 배치되었다. 무선할당은 모든 실험대상자 혹은 피험자가 모든 처치를 받을 확률이 동일함을 전제하기 때문에, 실험 집단들에 있어 피험자들의 특성이 서로 상쇄되어 동질성을 가질 수 있게 된다[16]. 실험집단(남12명, 여8명) 및 비교집단(남11명, 여9명) 중 단지 5명만이 인터넷 미디어를 통해 스크래치에 대해 들어본 적이 있었으며 대부분의 학생은 스크래치 프로그래밍에 대한 사전 경험이 존재하지 않았다.

본 연구에서 독립변인은 교수학습방법이고 종속변인은 학습자의 SW개념, SW구현능력 그리고 SW교육에 대한 태도였다. 두 집단은 같은 SW교육내용을 받았지만 유일한 차이점은 실험집단은 이야기를 활용한 반면 통제집단은 전통적인 방법의 교육을 받았다. 실험 처치는 동일 기간 및 시간 동안에 다른 교사에 의해 각각 실시되었다.

#### 3.2 교육프로그램

실험집단은 <Table 1>과 같이 이습우화를 활용한 SW교육을 실시하였다.

<Table 1> SW education program of experiment group

No	Category	Topics	SW concepts
1	Basics skill	Cat's bell	Sequences
2		Monkey and banana	Loops
3		Fox and crane	Conditionals
4		Three sons of farmer	Events
5		Bat's story	Operators
6		Greedy dog	Parallelism
7		Spring water	Data
8	Simulation	Farmer and lion	Sensing
9			
10	Game	Town musicians of Bremen	Function
11			
12	Art	Tiger's birthday party	Video sensing
13	Extension Tool	MakeyMakey	Controlling input devices
14			
15		Pico-Board	
16			
17		Lego Wedo	Problem solving
18			

초반부에는 순서, 반복, 조건, 연산, 변수, 이벤트 등의 내용을 포함하고 있는 이습우화 이야기를 듣고 SW의 기초 개념을 알고, 이야기 속의 한 장면을 스크래치로 응용해 볼 수 있도록 하였다. 3차시의 [조건]과 관련한 이습우화 자료로 『두루미는 부리가 뽀족하여 병에 담긴 음식을 받을 때는 맛있게 먹고, 접시에 담긴 음식을 받았을 때는 먹을 수가 없었다.』는 이습우화에서 두 조건에 따른 실행을 스크래치의 제어와 센서 블록을 이용하여 표현이 가능하였다.

중반부에서는 교사가 스토리텔링으로 제시한 문제를 듣고 해결을 위해 애니메이션, 시뮬레이션, 디지털아트 그리고 게임의 장르별 구현활동을 경험하였다. 끝으로 후반부에서는 전자보드와 로봇을 연동한 SW를 설계, 제작, 실행 및 개선하는 실제적인 문제해결활동을 경험하였다.

한편 비교집단은 전통적인 방식으로 실험집단과 동일한 학습내용을 처치 받았다. 초반부는 각 차시별로 지도교사가 5~10분 동안 순서, 반복, 조건, 이벤트, 연산, 병렬처리, 변수 등의 프로그래밍의 기본 개념을 안내한 후 스크래치 시범을 보이면 학생들은 각자 컴퓨터에서 따라 해보고 응용 발표해보는 활동을 하였다. 이후 게임, 시뮬레이션, 디지털아트, 애니메이션의 각 장르를 동일한 방법으로 실습하고 전자보드를 연결한 문제해결 활동을 경험하였다.

### 3.3 검사도구

SW교육 후 집단 간 CT능력 차이를 비교하기 위해 SW개념평가와 SW구현능력 평가를 실시하였다. 또한 SW교육에 대한 태도 측정을 위해 즐거움과 참여도를 평가하였다.

#### 3.3.1 SW개념 평가

스크래치 프로그래밍을 기초로 ‘알고리즘’, ‘논리사고’, ‘문제분해’의 세 영역의 문항을 연구자와 토요캠프 지도 교사가 각자 15문항씩 개발한 후 협의를 거쳐 25문항을 1차로 선정하였다. 이후 컴퓨터교육석사 소지자이면서 스크래치 프로그래밍 지도 경험이 있는 현장교사 3인의 문항 검토를 받아 수정 및 보완하였다. 내용타당도 검증 절차는 3명의 의견을 바탕으로 매우 그렇다 5, 그렇다 4, 보통이다 3, 그렇지 않다 2, 전혀 그렇지 않다 1로 하는 Likert 5단계 척도로 평가했으며, 평가 결과 평균이 3.0 미만인 경우 문항을 배제하였다. 또한 문항의 영역 및 난이도에 대한 의견이 다를 경우 합의를 통해 수정하였다. 최종적으로 다음 <Table 2>와 같이 알고리즘(10), 논리(6), 문제분해(4)의 총 20개 문항으로 구성되었다.

알고리즘은 순서, 반복, 조건, 변수, 이벤트, 연산의 SW기초개념 및 응용한 문제로 구성되었으며 논리는 프로그램의 실행 결과 예측, 논리적 오류를 찾고 교정 및 알고리즘 동작의 표현을 묻는 문항으로 측정하였다. 문제분해는 디버깅이 필요한 복잡한 코드 모음을 주고 작은 부분으로 쪼개어 해결할 수 있는지를 평가하였다. 문항은 <부록 1>에 첨부되었다.

<Table 2> SW concepts assessment contents

Category	No.	Contents	Level		
			High	Medium	Low
Algorithm	1	Sequence and loop			○
	2	Drawing a figure, Add-Block	○		
	3	Conditionals		○	
	4	Conditionals and Problem solving	○		
	5	Drawing a figure		○	
	6	Coordinate		○	
	7	Data, Performance of program	○		

Decomposition	8	Coordinate, Conditionals	○
	9	Coordinate, Conditionals	○
	10	Coordinate, Conditionals	○
	1	Relation, Decomposition	○
Logic	2	Debugging	○
	3	Random, Decomposition	○
	4	Decomposition	○
	1	Collision, Prediction	○
Logic	2	List, Prediction	○
	3	Drawing a figure, Prediction	○
	4	Drawing a figure, Prediction	○
	5	Data, Prediction	○
	6	Control, Prediction	○

#### 3.3.2 SW구현 능력

SW구현 능력 평가의 목적은 구현과정에서 지필평가에서 파악하지 못한 CT의 다른 요소를 종합적으로 평가하기 위함이다. 평가는 Malyn-Smith 외(2010)의 CT 평가 방식을 참조로 짧은 이야기 형식의 수행 과제를 제공한 후 SW구현 결과를 다음 조건에 따라 상, 중, 하의 3단계 수준으로 분류하였다[15]. 또한 과제에는 조건, 변수, 이벤트, 병렬화, 연산, 추가블록 등의 의무 사용에 관한 제약 조건이 추가되었다.

상 : 복잡한 솔루션 구현. 이야기 과제가 요구하는 모델링, 추상화에 대한 것을 이해하고 있다는 것을 보여주는 알고리즘 프로그램 구현. 제약 조건을 준수하고 프로그램의 실행에서 오류 없음

중 : 단순한 솔루션 구현, 코드를 읽을 수 있고 어디에 명령문을 배치해야 하는지 결정할 수 있음. 하지만 제약 조건을 지키지 못하거나 프로그램 실행의 오류가 있음

하 : 솔루션 구현 못함. 프로그램 미 작동

#### 3.3.3 SW교육에 대한 태도

초등학생의 SW교육에 대한 태도를 측정하기 위해 Wiebe(2003)의 컴퓨터과학태도 검사 도구를 수정 보완하였다[20]. 원래 총 57개의 문항으로 구성되었으나 일부는 프로그래밍보다는 일반적인 컴퓨터 과학과 관련되어 있거나 성별 차에 관한 문항으로 구성되었다. 그리고 검사지가 본래 대학생들 대상으로 개발되었기 때문

에 초등학생에게 그대로 적용하기에는 무리가 있었다. 따라서 프로그래밍과 관련한 초등학생들의 수준과 감정을 반영한 수정보완 작업을 거쳤으며 다음과 같이 자신감, 유용성, 동기영역의 총 9문항으로 선정하였다.

(\*)역문항

1. 나는 SW를 잘 배울 수 있다고 확신한다.
2. 나는 SW활동에서 더 많은 자신감을 갖고 있다.
3. 내가 열심히 노력하더라도 SW는 나에게 어려운 일이다(\*).
4. 나는 SW가 얼마나 유용한지 알기 때문에 공부한다.
5. 나의 미래 직업을 위해 SW를 확실히 배워야 할 필요가 있다.
6. 나는 내 전 생애를 통해 많은 방식으로 SW를 사용할 것이다.
7. 나는 SW를 만드는 것을 좋아한다.
8. 내가 SW문제를 즉시 해결할 수 없을 때 나는 해결할 때까지 꾸준히 노력한다.
9. 나는 바로 이해할 수 없는 SW문제에 도전감을 느낀다.

검사 문항은 5단계의 리커트 척도로 구성되었으며 결과 분석을 위해 ‘매우 동의’를 5점, ‘동의’를 4점, ‘보통’을 3점, ‘동의 않음’을 2점 그리고 ‘전혀 동의 않음’을 1점으로 하는 등간 척도로 변환하였으며 부정적인 문항인 3번 문항은 채점을 역으로 계산하였다. 따라서 본 검사지에서는 점수가 높을수록 문항에 대한 긍정적인 반응을 의미한다. 검사문항의 신뢰도 Cronbach’ a는 .708로 나타났다.

4. 연구 결과 및 해석

4.1 SW개념 평가 결과

스토리텔링기반 SW교육이 SW개념에 어떠한 변화를 주었는지 비교집단과 비교하여 t검증 결과는 다음 <Table 3>과 같다.

<Table 3> The result of SW concept assessment

Group	N	M	SD	t	df	Sig.(p)
Experimental	20	78.50	11.82	3.193	38	.003(**)
Comparative	20	65.50	13.85			

\*\* p<.01

실험집단은 78.50, 비교집단은 65.50로 13점의 평균 차이와 함께 95%의 유의수준에서 통계상 유의미한 차이를 보였다(p<.05). 즉 이습우화를 활용한 SW교육 방식이 전통적 교육방식보다 SW개념 습득에 더 효과적인 것으로 나타났다.

SW개념평가의 하위 영역별로 보면 알고리즘(p<.05), 문제분해(p>.05), 논리(p<.05) 중 주로 낮은 난이도 문항으로 구성된 문제분해 영역에서는 유의미한 차이가 나타나지 않았다. 실험 및 비교집단의 학습자들이 높은 오답을 보인 문항으로는 난이도가 중 또는 상 수준인 알고리즘의 A2, A4, A7, A8, A9, A10와 문제분해의 D1, D3 그리고 논리의 L2, L5문항이었다. 즉, 학습자는 스크립트의 단순 변환과 프로그램의 수행 결과를 묻는 비교적 간단한 사고를 묻는 문항에서는 높은 정답률을 보였으나 변수, 리스트, 좌표, 난수의 개념이 포함된 문항이나 두 가지 이상의 조건이 결합된 문항은 반대로 낮은 정답률을 나타냈다.

4.2 SW구현 능력 평가 결과

집단 간 SW구현 능력에서 유의미한 차이가 있는지를 검증하기 위해 평가 결과를 기초로 카이제곱(χ²) 검증을 실시하였으며 다음 <Table 4>와 같다.

<Table 4> The result of χ² test for SW implementation

Group	The result of Evaluation			Value	Sig.(p)
	High	Medium	Low		
Experimental	7(35%)	9(45%)	4(20%)	6.29	.043(*)
Comparative	2(10%)	7(35%)	11(55%)		

\* p<.05

카이제곱 통계량이 χ²=6.29, 점근유의확률이 .043이므로 스토리텔링기반 SW교육활동이 SW구현능력 평가에 더 효과적임을 알 수 있다.

실험집단은 주어진 조건을 포함하여 프로그램을 완

성한 학생이 7명(35%)인 데 비해 비교집단은 2명(10%)만이 오류 없이 프로그램을 완성하였다. 그리고 실험집단은 4명(20%)만이 과제를 해결하지 못한 반면 비교집단은 11명(55%)이 문제해결을 못하거나 제출한 프로그램이 제대로 동작되지 않았다.

SW구현 능력의 차이를 가져다 준 요인으로는 이야기를 표현하는 과정에서 다양한 시행착오를 통한 학습 경험, 이야기가 제공하는 풍부한 학습맥락을 생각해 볼 수 있다. 또한 학습자 주도의 학습 환경이 영향을 제공하였을 수도 있다. SW구현 과정에서 교사는 전통적인 SW구현 단계인 알고리즘 탐색-분석-설계-구현-디버깅의 과정을 유도하였지만 대부분의 실험집단의 학생들은 수행과제를 받은 후 시행착오를 겪으면서 완성해가는 Tinkering 활동을 하였다. 또한 실험집단의 학생들은 과제를 주었을 때 학생들이 직접 개발하는 것이 아니라 스크래치 커뮤니티의 프로젝트를 검색한 후 찾은 코드를 재사용하거나 동료의 작품을 모방하여 더 발전시키는 사례도 자주 관찰되었다. 즉, SW교육이 이루어지는 교실이나 온라인 공간에서의 커뮤니티를 통한 공유 활동이 구현능력 향상에 일정 부분 영향을 준 것으로 보인다.

### 4.3 SW교육에 대한 태도 결과

두 집단 모두 SW교육 활동에 즐겁게 참여하고 있는 것으로 나타났다. 하지만 실험집단에서 더 긍정적인 태도가 형성된 것으로 나타났다. SW교육에 대한 집단별 t 검증 결과는 다음 <Table 5>와 같다.

<Table 5> The result of attitude for SW education

Group	N	M	SD	t	df	Sig.(p)
Experimental	20	37.85	3.18	4.11	38	.000(**)
Comparative	20	32.70	4.61			

\*\*  $p < .01$

실험집단은 평균  $M=37.85$ , 표준편차  $SD=3.18$ 로 나타났으며, 비교집단은 평균  $M=32.70$ , 표준편차  $SD=4.61$ , 통계적 분석 결과  $p=.000$ 으로  $\alpha=.05$  수준에서 유의미한 차이가 있는 것으로 나타났다. 즉, 스토리텔링이 SW교육 태도 형성에 긍정적인 영향을 주었다는 것을 의미한다.

### 5. 결론 및 제언

우리는 SW교육의 필요성을 주장할 때 고차원의 사고능력 함양을 근거로 제시하고 있다. 하지만 학습자나 학부모 관점에서 볼 때 ‘굳이 SW를 활용해야 하는가?’ 즉 ‘왜 SW교육을 배워야 하는가?’에 대한 설명으로는 1% 부족하다. 즉, SW교육에 대한 수요자의 자발적인 지적 호기심의 유도가 선결되어야 한다.

이런 측면에서 본 연구는 학생들에게 친숙한 이습우화를 SW교육에 활용하였다. 연구결과 스토리텔링을 활용한 실험집단은 전통적 교육방식을 택한 비교집단과 비교하여 SW개념, SW구현능력 그리고 SW교육태도에서 긍정적인 결과를 보였다. 이러한 결과는 프로그래밍에 대한 학생들의 태도와 성취결과에는 긍정적인 상관관계가 존재한다는 기존 연구결과와도 일치하는 것으로 [3] 향후 CT를 함양하기 위해서는 SW교육에 대한 좋은 교수-학습환경이 선결되어야 함을 시사한다. 하지만 본 연구는 몇 가지 제약을 갖고 있다.

첫째, 스토리텔링의 교육적 효과에 관한 연구를 양적으로만 비교했다는 점에서 한계를 갖는다. 향후 연구에서는 이야기 주제를 주면 학생들이 어떻게 SW교육활동을 하는지 수업관찰, 분석, 인터뷰 등을 통해 CT를 구체적으로 어떻게 적용하고 발전시켜 나가는지에 관한 질적 연구가 후속될 필요가 있다.

둘째, CT 측정을 위해 SW개념과 SW구현능력을 평가하였는데, SW개념은 알고리즘, 문제분해, 논리의 세 영역으로만 제한하였다. 향후 추상화, 패턴, 자동화, 일반화가 포함된 검사를 실시할 필요가 있다. 또한 SW구현능력을 5단계로 세분화하여 평가가 이루어진다면 구현능력을 보다 객관적으로 검증할 수 있을 것으로 사료된다.

셋째, 프로그래밍 교육에서 성별 차이 분석은 중요한 과제로 SW개념 및 구현능력에서 스토리텔링이 어떠한 결과를 제공하는지 연구할 필요가 있다.

### 참고문헌

[1] Allan, W., Coulter, B., Denner, J., Erickson, J., Lee, I., Malyn-Smith, J., Martin, F. (2010). Computational thinking for youth. White Paper for the ITEST

- Small Working Group on Computational Thinking (CT).
- [2] Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *ACM Inroads*, 2(1), 48- 54.
- [3] Baser, M. (2013). Attitude, Gender and Achievement in Computer Programming. *Online Submission*, 14(2), 248-255.
- [4] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.
- [5] Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- [6] CSTA (2011). CSTA Computer Science Standards Revised 2011, The CSTA Standards Task Force.
- [7] Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, referenced in <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- [8] Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing computational thinking in the classroom: a framework.
- [9] Department for Education (2014). The national curriculum in England, Retrieved from <https://www.gov.uk/government/publications/national-curriculum-in-england-primary-curriculum>
- [10] Fields, D. A., Searle, K. A., Kafai, Y. B., & Min, H. S. (2012). Debuggems to assess student learning in e-textiles. In Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education. New York, NY: ACM Press.
- [11] Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245-255.
- [12] Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys(CSUR)*, 37(2), 83-137.
- [13] Kim, K. Y., Song, J. B. & Lee, T. W. (2009). Effect of Digital Storytelling based Programming Education on Motivation and Achievement of Students in Elementary school. *Journal of Korea Society of Computer and Information*, 14(1), 47-55.
- [14] Lee, J. I, Sung, Y. H (2011). Development of Robot Programming Education System for Children based on Storytelling. *Journal of Korean Information Education*, 15(2), 295-305.
- [15] Malyn-Smith, J., Coulter, B., Denner, J., Lee, I., Stiles, J. & Werner, L. (2010). Computational Thinking in K-12: Defining the Space. In D. Gibson & B. Dodge (Eds.), Proceedings of Society for Information Technology & Teacher Education International Conference 2010(pp.3479-3484). Chesapeake, VA: Association for the Advancement of Computing in Education (AACE).
- [16] Michael, K. Y. (2001). The Effect of a Computer Simulation Activity Versus a Hands-on Activity on Product Creativity. *Journal of Technology Education*, 13, 31-43.
- [17] Park, J. H. (2014). A Study on Digital Storytelling Based Programming Education. *Journal of the Korea Society of Computer and Information*, 19(5), 119-128.
- [18] SPRI (2014). Software Centered Society: The Meaning and Direction of Reaction, SPRI Issue Report 2014-003.
- [19] Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012, February). The fairy performance assessment: Measuring computational thinking in middle school. In Proceedings of the 43rd ACM technical symposium on Computer Science

Education (pp. 215-220). ACM.

- [20] Wiebe, E. N., Williams, L., Yang, K., & Miller, C. (2003). Computer Science Attitude Survey, (Report No.: NCSU CSC TR-2003-1) Dept. of Computer Science, NC State University, Raleigh, NC.
- [21] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

저자소개



박 정 호

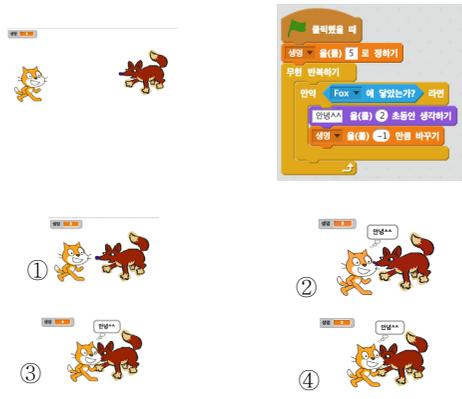
1997 서울교육대학교 과학교육학  
과(교육학학사)  
2008 한국교원대학교 컴퓨터교육  
과(교육학박사)  
2013 Tufts University CEEO  
Research Scholar  
2015~현재 서울교육대학교 교육  
전문대학원 겸임교수, 가수  
초등학교 교사  
관심분야: 컴퓨터교육, SW교육  
e-mail: jhpark0154@gmail.com

[부록-1] SW개념평가 문항

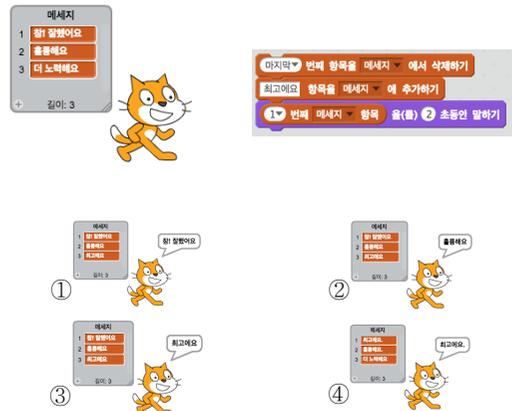
[A1] 다음 스크립트를 간단히 줄여보세요.



[L1] 다음 프로그램을 실행하였을 때 결과는 무엇인가요?



[L2] 다음 리스트의 프로그램을 실행하였을 때 결과는 무엇인가요?



[A2] 크기가 150인 정사각형을 그리려고 합니다. 다음 스크립트를 완성하세요. (      ,      ,      )



[D2] 고양이가 항상 쥐를 쫓아가는 프로그램입니다. 오류를 찾고 해결방안을 쓰세요.



오류 : \_\_\_\_\_

해결방안 : \_\_\_\_\_

[D1] 시계의 시침을 동작하는 스크립트 완성을 위한 블록의 값은 무엇일까요? (      ,      )



[L4] 다음 프로그램의 실행 결과는 무엇인가요?



[L3] 다음 프로그램의 실행 결과는 무엇일까요?



[A3] 오른쪽과 같이 조건문 안에 조건문을 넣을 수 있습니다. 비어있는 공간에 어떤 블록을 넣으면, 2개의 조건문을 1개의 조건문으로 합칠 수 있습니다. 각각 어떤 블록을 넣으면 될까요?



[A4] 농부는 강아지, 염소, 양배추 세 가지를 강 건너편으로 옮기려 합니다. 농부는 배를 한 척 갖고 있으며 한 번에 두 가지 물건을 옮길 수 있습니다. 하지만 개와 염소가 함께 남겨진다면 개는 염소를 먹게 되고 염소와 양배추가 함께 남겨진다면 염소는 양배추를 먹게 됩니다. 세 가지를 안전하게 옮기기 위해 총 몇 번 강을 건너야 할까요? (      )

[D3] 다음 스크립트를 통해 고양이가 움직일 수 있는 거리의 경우를 써보세요. (      )



[L5] '야옹'은 몇 번 소리가 나며 스크립트가 종료될 때 횟수 변수는 값은 얼마일까요?  
소리 : (      ),  
횟수 : (      )



[A5] 한 번의 길이가 300인 정사각형을 그리려고 합니다. 다음의 스크립트를 완성하세요.

① \_\_\_\_\_

② \_\_\_\_\_

③ \_\_\_\_\_

[A7] 아래 스크립트가 종료된 후 두 변수 값은 각각 무엇인가요? ( )

① counter 6 limit 7

② counter 7 limit 7

③ counter 8 limit 7

④ counter 7 limit 8

[D4] 고양이를 90씩 회전시키려합니다. 하지만 고양이는 회전하지 않습니다. 어떻게 하면 회전시킬 수 있을까요? ( )

[A8-A10] 다음 프로그램이 실행되었다는 것을 고려하여 문제에 대한 답을 하시오.

[L6] 다음 스크립트를 실행하였을 때 '아웅'은 몇 번 들을 수 있을까요? ( )

① 4번    ② 5번

③ 6번    ④ 7번

[A8] 만약 스프라이트가 x=-25 y=-25의 위치에서 시작하고 'c'를 누르면 어떻게 되는가? ( )

- ① 스프라이트가 x:-25 y:25로 이동하고 y위치는 0보다 25위라고 말한다.
- ② 스프라이트가 x:25 y:25로 이동하고 y위치는 0보다 25위라고 말한다.
- ③ 스프라이트가 x:25 y:-25로 이동하고 y위치는 0보다 25아래라고 말한다.
- ④ 스프라이트가 x:25 y:25로 이동하고 y위치는 0보다 25아래라고 말한다.

[A6] 왼쪽 키를 누르면 왼쪽으로 오른쪽 키를 누르면 오른쪽으로 움직이는 고양이 스크립트입니다. 고양이의 x좌표가 0보다 크면 오른쪽에 있다고 말하고 0보다 적으면 왼쪽에 있다고 말하게 하려합니다. 하지만 고양이는 아무런 말도 하지 않습니다. 어떻게 해결해야 할까요? ( )

[A9] 스프라이트가 x:-25 y:-25에서 시작하고 'a', 'b', 'c' 그리고 'd'키를 누르면 어떻게 되는가? ( )

- ① 스프라이트가 x:-25 y:-25로 이동하고 y위치는 0보다 25아래라고 말한다.
- ② 스프라이트가 x:25 y:25로 이동하고 y위치는 0보다 25위라고 말한다.
- ③ 스프라이트가 x:25 y:75로 이동하고 y위치는 0보다 75위라고 말한다.
- ④ 스프라이트가 x:75 y:25로 이동하고 y위치는 0보다 75위라고 말한다.

[A10] 스프라이트가  $x:-25$   $y:-25$ 에서 시작하고 'a', 'c', 그리고 'e'키를 누르면 어떻게 되는가? ( )

- ① 스프라이트가  $x:25$   $y:-25$ 로 이동하고 y위치는 0보다 25위라고 말한다.
- ② 스프라이트가  $x:75$   $y:25$ 로 이동하고 y위치는 0보다 25위라고 말한다.
- ③ 스프라이트가  $x:75$   $y:-25$ 로 이동하고 y위치는 0보다 25아래라고 말한다.
- ④ 스프라이트가  $x:25$   $y:-25$ 로 이동하고 y위치는 0보다 25아래라고 말한다.