Journal of The Korean Association of Information Education Vol. 19, No. 2, June 2015, pp. 243-252

Computational Thinking 기반의 초등학교 동아리 활동용 프로그래밍 교육 교재의 개발

정인기

춘천교육대학교 컴퓨터교육과

요 약

2019학년도부터 초등학교에서 소프트웨어 교육이 실시될 예정이다. 초등학교 소프트웨어 교육의 주된 내용 중의 하나는 학생들이 프로그래밍을 체험하는 것이다. 프로그래밍에 대하여 관심을 보이는 학생들에게는 보다 높은 수준의 프로그래밍 교육이 필요하며 동아리 활동을 통하여 시행할 수 있다. 그러나 초등학교에서의 동아리 활동을 위한 프로그래밍 교재는 많지 않다. 따라서 본 연구에서는 초등학교의 동아리 활동을 위한 프로그래밍 교재를 개발하였다. 따라 하기 형태의 매뉴얼 방식을 지양하고 학생들이 문제를 이해하고, 문제 분할 및 추상화과정을 따라 설계하며 이를 프로그램으로 작성할 수 있도록 하여 자연스럽게 Computational Thinking 능력을 배양하도록 하였으며 시범 수업을 통하여 개발한 교재가 초등학생들에게 적절한 내용과 수준임을 입증하였다. 따라서 본 연구에서 제시하는 교재 집필 방법들이 초등학교의 소프트웨어 교육에 기여할 것으로 기대한다.

키워드: 초등학교 프로그래밍 교육, 프로그래밍 교재, 스크래치 프로그래밍

Development of Materials for Programming Education based on Computational Thinking for Club Activities of Elementary School

Inkee Jeong

Dept. of Computer Education, ChunCheon National University of Education

ABSTRACT

The software education to elementary students will be conducted from 2019. One of highlights of software education is a programming experience. It requires a higher level of programming education to students that are interested in programming. This problem can be solved by the club activities. But the materials for programming education for elementary students is not much. Therefore, we developed a programming material for club activities of the elementary school. We did not develop it as a programming manual. The students can understand a problem, can design through decomposition and abstraction processes, and can write a program when they are learning with this material. As a result, we expect that they can enhance their computational thinking abilities. We proved that our material is suitable for elementary students through a demonstration class. Therefore, we expect that our development methodologies for the material for programming education will contribute to develop a material for programming education.

Keywords: Programming Education for Elementary Students, Programming Material, Scratch Programming

논문투고: 2015-05-30 논문심사: 2015-06-01 심사완료: 2015-06-23

1. 서론

21C 지식 정보화 사회는 창의적이고 문제해결력을 갖춘 인적자원을 필요로 하고 국가 경쟁력은 우수한 인재를 육성하는 것에서 출발한다고 볼 수 있다. 따라서세계 각국은 우수 인력 양성을 중요한 국가적 과제로여기며 다양한 교육적 노력을 기울이고 있다. 즉 인재양성은 국가 차원에서 많은 관심을 갖게 되었으며 과학적이고 체계적으로 해야 한다는 방향이 설득력을 얻고있다[10].

특히 정보 사회의 인재 양성을 위해서 소프트웨어 교육이 필수적이며 소프트웨어 교육 내용에는 많은 분야들이 있으나 알고리즘, 프로그래밍 그리고 Computational Thinking 등이 주요 내용 분야들이라고 할 수 있다.

알고리즘은 초기 상태로부터 목표 상태를 만들어 내기 위한 일련의 단계를 생성하는 것이므로 일반적인 문제 해결 과정[12][17]과 유사하기 때문에 알고리즘 학습을 통해 문제 해결 절차를 학습할 수 있을 것으로 기대할 수 있다. 이러한 알고리즘 학습은 문제 해결력 향상에 영향을 미칠 수 있을 것으로 기대되며, 실험에 의해그 효과를 증명하고 있다[2][8].

또한 프로그래밍 과정은 문제 해결을 위해 적합한 추상적 개념을 선택하고 구성하기 위한 추상화 능력과 추상적 개념을 자동화하기 위해 적합한 컴퓨팅 도구를 선택하고 사용하기 위한 자동화 능력을 요구한다[22]. 이러한 인지 능력이 다른 학문 분야에서 다루는 문제해결기술이나 추론기술과 다른 점은 보다 일반적인 문제해결상황에 적용할 수 있는 전략적 지식을 생성하고, 컴퓨팅 장치를 통해 문제해결과정을 실제로 구현하고 자동화한다는 측면에서 가장 정확하고 명확한 해법을 제시할 수 있다는 점이다[13].

학습자들이 프로그래밍을 배워야 하는 이유는 학습에 대한 피아제(Piaget)식 사고와 관련이 있다. 로고의 개발자인 Seymour Papert는 프로그래밍 과정에서 학습자들은 자신의 생각을 프로그램으로 표현하고 프로그램이 동작할 때까지 디버깅을 하면서 자신의 생각에 대하여 생각한다고 하였으며[18][23], Guzdial은 "학습자는 자신의 프로그램을 디버깅하면서 학습자들은 자신의 생각을 디버깅한다. 프로그래밍은 자신의 생각에 대하여생각하면서 디버깅과 계획과 같은 고차원적인 사고 능

력을 얻을 수 있다."라고 하였다[6][23].

뿐만 아니라 Computational Thinking의 핵심적인 의미는 인간이 실생활에서 직면할 수 있는 다양하고 복잡한 문제를 어떻게 해결할 것인지를 절차적으로 사고하고, 문제의 해결 과정을 컴퓨팅 기기가 제공하는 강력한 능력들을 통해 효과적이고 효율적으로 해결하고자하는 종합적인 사고과정이라고 할 수 있다.

기존에는 컴퓨팅 기기를 단순히 반복적이고 지루한 작업 을 대신하는 일반적인 기기로 인식했다면, Computational Thinking의 개념에서는 컴퓨팅 시스템이 가지고 있는 다양한 능력을 인간의 사고 능력과 통합하여 새로운 형 태의 인지 구조를 활용하는 것이라 할 수 있다. 즉, 복 잡하고 다양한 문제를 해결하기 위해 필요한 자료를 수 집 · 추출 · 분석하여 문제의 해결 모델을 구축하고, 이 러한 모델을 컴퓨터가 이해할 수 있는 언어로 구현하여 문제의 해결 방법을 발견하는 것이다. Computational Thinking을 문제 해결 과정에 활용하게 된다면 복잡한 대량의 자료들을 빠르게 처리할 수 있도록 하며, 컴퓨 터의 힘을 빌려 인간의 사고 능력 이상의 것들을 수행 하도록 함으로써 문제 해결에 필요한 사람의 인지 구조 를 확장시켜주는 역할을 한다. Computational Thinking 은 컴퓨팅 기술이 사회의 근간이 되고 이를 활용하여 복잡한 문제를 해결하는 능력이 필요한 21세기에 학습 자가 갖추어야 할 핵심 역량으로 강조되고 있으며, 읽 기, 쓰기, 셈하기(3R)의 기초 문해력과 같이 모든 사람 이 반드시 알아야 할 기초 소양으로 인식되고 있다 [15][22].

특히 ISTE에서는 Computational Thinking의 요소로 자료수집, 자료 분석, 자료 표현, 문제 분할, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화 등을 제시하고 있으며 이를 학생들에게 교육할 것을 제안하고 있다[15].

지식정보화 사회라고 불리는 급변하는 사회 속에서 학교 교육은 학생들이 직면하게 될 미래사회를 준비하는 방향으로 나아갈 필요가 있다. 개개인의 건강한 삶과 더불어 사회의 지속적인 발전에 필요한 자질을 함양시키기 위한 교육이 이루어져야 한다. 즉, 학교 교육은 미래 사회의 주역이 될 학생들에게 사회적 변화에 적응하고 준비하는 데 필요한 교육을 제공해야 할 필요가 있다[11]. 이에 따라 우리나라도 2019학년도부터 초등학

교 5~6학년에서 실과 교과 내에서 소프트웨어 교육을 실시할 예정이다. 이러한 시점에서 현재 시급한 것은 교사들에 대한 연수와 다양한 교재의 개발이라고 할 수 있다. 새로운 교육 내용에 대한 교사 연수는 반드시 이루어져야 할 것이며, 이와 함께 다양한 내용과 수준의 교재 개발도 필수적으로 이루어져야 소프트웨어 교육을 성공시킬 수 있을 것이다.

특히 제대로 된 소프트웨어 교육을 위해서 17차시는 매우 적은 시간이다. 그러나 시수를 좀 더 확보하기 어렵다면 현실적으로 가능한 다른 방안을 모색하는 것도시급한 일이다. 여러 방법 중에서 소프트웨어 개발 등에 관심을 보이는 학생들을 대상으로 동아리 활동을 시행하는 것도 하나의 방안이 될 수 있을 것이다. 그런데 초등학교에서 동아리를 운영하고 싶어도 활용할 수 있는 교재가 개발되어 않으면 많은 어려움을 겪을 수 있기 때문에 다양한 교재의 개발이 필수적이라 할 수 있다. 따라서 본 연구에서는 초등학교 동아리 활동에서 활용할 수 있는 프로그래밍 교재를 개발하였다.

2. 초등학교 프로그래밍 교육 현황

프로그래밍은 학습자 주도의 능동적인 학습을 가능하게 하며 즉각적인 피드백이 가능한 상징적인 체계 속에서 구체적이고 엄격한 사고와 정확한 표현을 요구하며, 가설을 세우고 검증할 수 있는 학습 환경을 제공한다. 학습자는 프로그래밍 과정에서 자신이 학습의 주체가 되어 주어진 과제를 이해 · 분석하고 과제 수행을 위한 방법을 찾으며 그 방법을 목적한 대로 실행하고 발생되는 오류를 주체적으로 해결하는 과정을 경험을 통해 전통적 교과 학습을 자기주도적으로 이끌러 갈 수있는 능력을 배양할 수 있다[3][16].

즉 프로그래밍 교육의 학습 목표는 프로그래밍에 대한 기초 지식을 습득하고 프로그래밍 언어의 개념과 활용 방법 및 프로그래밍 기법을 익혀 정보화 사회에서 필요로 하는 여러 가지 문제를 해결하고 자료 처리를 위한 프로그램을 작성할 수 있는 능력과 태도를 기르는데 있다. 이러한 프로그래밍 교육이 미래의 주역이 될학생들에게 필요한 교육이지만 실제 프로그래밍 교육은교사에게는 어려운 프로그래밍 언어와 문법적 지식의

습득을 가르쳐야 한다는 부담감을 가져다주며 학생들에 게는 프로그래밍 수업을 접해 볼 기회가 적어서 어렵고 재미없는 과목이라는 부정적인 생각을 심어줄 수 있다. 그나마 실시되고 있는 프로그래밍 수업이 단지 문제해 결학습에만 치우쳐 프로그래밍을 학습하는 학생들에게 오개념을 형성시킬 가능성이 있고 제한된 학습 시간에 적용하기에는 여러 가지 어려움이 있다[3].

이러한 문제점을 개선하기 위한 노력으로 초등학생을 위한 프로그래밍 교육에서 초등학생의 수준에 맞는 교육용 프로그래밍 언어가 많이 사용되고 있다. 교육용 프로그래밍 언어는 프로그래밍 교육에서 학습자들이 가지는 다양한 인지적 부담을 줄여주고, 동기를 부여할 수 있도록 교육적인 목적을 가지고 개발된 프로그래밍 언어를 말한다[20].

교육용 프로그래밍 언어는 사용하기 쉬운 인터페이스와 다양한 멀티미디어 기능을 제공한다. 이는 프로그래밍 언어 사용법 학습이나 문법적 오류로 인해 부가될 수있는 과도한 인지적 부담을 감소시킴으로써 초등학교 학습자도 쉽게 사진의 생각을 컴퓨터로 표현할 수 있도록도움을 준다[1]. 이처럼 교육용 프로그래밍 언어를 사용한 프로그래밍 교육은 학습자의 내적 동기를 유발하며일반적인 프로그래밍 교육보자 자기주도적 학습 능력을향상시키는 데 효과적인 것으로 나타났다[21][25].

프로그래밍 도구는 개발자가 프로그래밍을 할 수 있는 도구를 말하며, 학생들의 경우에는 프로그래밍 도구가 프로그래밍 교육의 효과에 많은 영향을 끼친다. 따라서 교육용 프로그래밍 도구는 비주얼 도구의 형태를 가지는 경우가 많으며 교육용 프로그래밍 도구를 프로그래밍 방식에 따라 분류하면 < Table 1>과 같다[7].

< Table 1> Classification According to the Programming Style

Programming Tool	Programming with Code Blocks	Graphic Output & Type Coding	etc	Support High Level Language
Scratch Jr	0			
Scratch	0			
Alice	0			0
App Inventor	0			
Snap!	0			
Entry	0			0
Hopscotch	0			
KODU	0			

Pico Block	0			
S4A, S2A	0			
Mindstorm	0			0
Logo		0		
Jeroo		0		0
RUPLE		0		0
Green foot		0		0
etoys			0	
physical etoys			0	
Dolittle			0	

<Table 1>과 같이 많은 교육용 프로그래밍 도구 중에서 초등학생용으로 가장 많이 사용되는 것은 스크래치이다. 스크래치 프로그래밍 도구가 많이 사용되고 있다. 2007년 MIT Mdeia Lab의 연구팀에 의해 개발된스크래치는 Mitchel Resnick에 의해 디자인되었으며, Squeak Etoys에 그 기초를 두고 있다. 스크래치는 프로그래밍 교육을 위해 개발된 것으로 객체지향 언어이고, 스크래치 홈페이지를 통해 무료로 배포된다. 스크래치는 어린 아이들에게 프로그래밍의 기본 개념과 알고리즘을 가르치기 위해 개발되었으며, 다양한 멀티미디어지원을 통해 쉽게 게임이나 애니메이션 등을 만들 수있다. 문법 위주의 교육보다는 프로그램의 구조를 익히는 것과 논리적인 문제에 초점을 맞췄다. 때문에 프로그래밍 초보자의 입문 과정으로 적합하다[3].

스크래치는 최근 스크래치 2.0으로 개편되며, 새로운 기능들이 추가되었다. 기존과 달리 스크래치 2.0에서는 웹 브라우저에서 바로 프로젝트를 만들고, 편집하고, 실 행되는 것을 확인할 수 있다. 스크래치 2.0의 추가된 기 능 중 가장 큰 특징은 온라인 커뮤니티에 공유된 다른 사람들의 프로젝트가 어떻게 동작되는지 코드를 직접 확인해 볼 수 있다는 것이다. 이를 통해 다른 사람이 공 유한 프로젝트로부터 배경이나 스프라이트, 스크립트를 자신의 개인 저장소로 옮겨 담아 자신의 프로젝트에 활 용할 수 있다. 또한 다른 사람의 프로젝트를 내 프로젝 트로 저장하여 변경해 볼 수 있으며, 변경된 프로젝트 를 다시 공유하면 최초로 프로젝트를 만든 사람의 프로 젝트와 연결된다. 이와 같이 스크래치 2.0에서는 다른 사람들과 상호작용할 수 있는 공간과 협력적인 작업을 할 수 있는 기회를 제공하고, 아이디어를 공유하고 이 를 수정하고 보완하며 학생들의 학습을 지원할 수 있는 기능이 확대되었다고 할 수 있다[19].

따라서 본 연구에서는 초등학생들에 대한 프로그래 밍 교육 현황 등을 고려하여 스크래치 프로그래밍을 위 한 교재를 개발하였다.

3. 초등학생용 프로그래밍 교재의 문제점

초등학생들에게 프로그래밍을 가르치는 이유는 Computational Thinking 능력을 배양하기 위한 것이다. 그런데 기존의 프로그래밍 교재들은 학생들의 Computational Thinking 교육면에서 보면 다음과 같은 문제점들을 포함하고 있다.

3.1 단순한 따라 하기

대부분의 스크래치 프로그래밍 교재들은 문제를 제시하고 그 문제를 해결하는 프로그램을 제시한다. 이때 대부분의 학생들은 제시한 프로그램을 그대로 따라하게 된다. 이러한 방식은 학생들이 스크래치라는 프로그래밍 도구의 사용법을 익히는 데는 도움이 되지만 문제를해결하는 방법을 습득하는 데 도움을 주지 못할 수도있다. 따라서 code.org와 같이 온라인 튜토리얼에서 제공하는 것처럼 문제를 해결하도록 유도하고 다음 단계로 진행하도록 하는 방법에 비하여 효과가 감소될 수있다. 이것은 서책형의 단점이라고 볼 수 있으며 수업에 활용할 때 선생님의 역할이 줄어들게 되는 현상을 보일 수도 있다. 따라서 학생들은 문제 해결 방법을 탐색하려는 노력을 덜 하게 되며 이는 학생들에게 Computational Thinking을 교육하기 어렵게 한다.

3.2 문제 해결 방법 제언의 부재

스크래치 프로그래밍 교재들의 대부분은 따라 하기에 치중한 나머지 어떻게 문제를 해결할 것인가에 대한 방법을 보여주지 못하는 경우가 많다. 즉 문제 해결 과정은 생략하고 단순히 프로그램만을 보여주어 학생들이 문제해결 방법에 대하여 이해하기 어려운 경우가 많다. 따라서 학생들이 주도적으로 자료들을 수집, 분석, 표현하고이를 바탕으로 문제 해결 방법을 찾아가는 것을 중시하는 Computational Thinking을 교육하기 어렵게 된다.

3.3 명령 유형별 매뉴얼식 구성

많은 스크래치 프로그래밍 교재들은 코드 블록의 종류에 따라 교재를 구성하고 있다. 이러한 방식은 스크래치 전체의 구조를 익힐 수 있어 프로그래밍 경험이 있거나 성인의 경우에는 효과적인 방법이 될 수 있으나프로그래밍 경험이 없는 학생들의 경우에는 프로그래밍에 대한 흥미를 유발할 수 없는 단점을 가지고 있다. 이는 학생들의 참여를 어렵게 함으로써 Computational Thinking 교육 자체가 힘들어지게 한다.

4. 초등학생용 프로그래밍 교재 개발

4.1 교재 개발 방향

본 교재는 학생들에게 Computational Thinking을 소개하고 교육하는데 있다. 따라서 Computational Thinking을 교육하기 위하여 구현한 방법은 다음과 같다.

(1) 줄거리 기반 프로그래밍

Computational Thinking의 대표적인 절차는 추상화이다. 추상화는 ISTE에서와 같이 문제 분할과 반대되는 방향으로써의 설계 절차를 의미하기도 하지만 영국의경우와 같이 모델링, 분할 및 일반화를 합친 개념으로 사용되기도 한다. 추상화 과정은 설계 과정으로서 결과물인 프로그램에 반드시 반영되어 표현되어야 하는 것은 아니다. 그러나 학생들의 경우에 이를 표현하지 않고사고의 과정으로서만 교육하는 경우에 소기의 목적을달성할 수 없다. 표현하지 않는 경우에 학생들이 어떤사고를 하였는지 알 수 없기 때문이다. 따라서 문제 분할과 일반화 과정을 프로그램에서도 표현하도록 하는 것이 교육 목적을 달성할 수 있는 좋은 방법이 될 것이다. 결국 이와 같은 사고 절차는 줄거리를 먼저 구성하고, 이를 상세화하는 것으로 표현할 수 있으며, 스크래치에서는 추가 블록 기능을 사용하면 구현할 수 있다.

스크래치의 추가 블록은 일반 프로그래밍 언어의 부 프로그램에 해당하는 것으로 드라이버(driver)나 스텁 (stub)의 역할을 수행할 수 있다. 부프로그램을 가동시 키는 드라이버나 완성된 부프로그램을 흉내 내는 스텁은 원래 프로그램을 테스트하기 위한 것이지만[14] 본교재에서는 분할된 문제를 표현하기 위해 사용하였으며, 처음에는 완성되지 않은 형태로 문제 해결 방법 혹은 스토리텔링의 줄거리를 추가 블록으로 표현한 후에이를 상세하게 프로그래밍하는 방법을 사용하였다.

(2) 단계적 따라하기

인지적 도제는 학교교육 제도가 갖추어지지 않았던 과거의 지식이나 기술의 전수 방법으로 사용되었던 전통적 도제 방법을 현 사회에서 요구하는 형태로 변형한 교수-학습 방법이다. 눈으로 확인할 수 있는 외형적 지식이나 기능의 전수를 도모했던 도제 방법과는 다르게 인지적 도제는 과제 관련 지식 습득과 함께 사고력, 문제해결력 등과 같은 고차원적 인지 기능의 신장을 도모할 수 있는 교수-학습 방법이다[5][9][24].

인지적 도제 기반의 교수-학습 전략으로는 모델링 (modeling), 코칭(coaching), 스캐폴딩(scaffolding), 명료화 (articulating), 반성적 사고(reflecting)와 탐구(exploring)가 있다[4][24].

본 교재에서는 교사가 시범을 보이고, 학생은 교사의 시범을 관찰하면서 과제를 해결하는 과정에 대해 이해 하도록 하는 모델링과 학생이 과제의 일부를 수행하도록 하고, 교사는 그 활동을 지원하여 학생 스스로 과제를 수행할 수 있도록 돕고 학생의 과제 수행 능력이 향상되면 점차적으로 교사의 도움 제공 횟수를 줄이는 스케폴딩 방법을 사용하였다. 즉, 학생들이 본 교재에서 제시하는 문제 해결 방법을 따라하면서 프로그래밍 방법을 익힐 수 있도록 하였으며, 처음에는 모든 프로그램을 다 보여주면서 따라하기를 하도록 하였지만 다음에는 프로그램 내의 일부 블록을 비워서 학생들이 설명을 보고 채워 넣을 수 있도록 하였다. 그 후에는 추상화에 해당하는 블록의 이름을 비워두어 학생들이 블록의이름을 작성할 수 있게 하여 따라 하기를 단계적으로줄일 수 있도록 하였다.

(3) 문제 해결 체험을 위한 단원 구성

프로그래밍에 대한 단원인 Ⅱ단원과 Ⅲ단원의 단원 구성은 다음과 같다.

- -단원 개요
- -학습 목표
- -도전 과제 제시
- -과제 해결
 - 처음 상태를 만들어요.
 - 프로그램에서 사용할 변수를 만들어요.
 - 스프라이트 움직임의 줄거리를 구성해요.
 - 스프라이트들의 줄거리 블록들을 상세하게 프로 그래밍해요.
- -실행하여 확인해 보세요.
- -더 해보기
 - •문제 제시
 - 과제 해결
 - 실행하여 확인해 보세요.
- -정리해 보기
- -생각해 보기

특히, 주요 내용이 되는 과제 해결 부분에서는 학생들 이 프로그래밍을 체계적으로 접근할 수 있도록 하기 위 하여 일정한 체계를 제시하였다. 즉, 프로그래밍을 하는 구조로서 "처음 상태를 만들어요", "프로그래밍에서 사 용할 변수를 만들어요", "스프라이트 움직임의 줄거리를 구성해요". "스프라이트들의 줄거리 블록들을 상세하게 프로그래밍해요"의 4단계로 구성하였다. 이와 같은 4단 계의 프로그래밍 과정은 스크래치와 같은 블록형 프로 그래밍 도구에 최적화된 것이다. 먼저, 과제를 해결하기 위한 초기 상태를 만들고, 변수를 정의한다. 변수는 필 요한 경우에 만들도록 한다. 다음에는 문제 분할 및 추 상화 단계로써 줄거리를 구성한다. 본 교재에서는 줄거 리를 단순히 설계의 과정으로만 국한하는 것이 아니라 부프로그램(예, 스크래치의 경우에는 "추가 블록")의 형 태로 구현하도록 제시하였다. 이와 같이 하면 학생들의 문제 분할 및 추상화를 통한 설계 과정을 표현할 수 있 고, 프로그램에 대한 주석 역할도 할 수 있게 된다. 다음 에는 작성한 줄거리를 상세하게 프로그래밍한다.

4.2 대상

본 교재는 초등학교 5~6학년의 동아리 활동용으로 제작되었다. 따라서 프로그래밍에 대한 개념을 파악하 고 있는 교사가 필요하며, 프로그래밍에 관심을 가지고 있는 학생들을 대상으로 하였다. 본 교재를 사용하기 위해 필요한 선수 지식은 필요하지 않지만 프로그래밍 에 관심 있는 학생들을 위하여 도전할 수 있는 주제들 을 선정하였다.

4.3 교재 내용

본 교재는 매뉴얼식의 구성을 지양하고 학생들이 홍미를 가질 수 있도록 주제별로 구성하였다. 특히 주제의 특성에 따라 성별 차이를 보이는 것을 고려하여 게임과 같은 문제 해결 관련 주제와 디지털 스토리텔링에관한 주제를 모두 수록함으로써 프로그래밍에 대한 흥미도에 있어서의 성별 차이를 극복하려고 하였다.

이와 같이 선정된 주제로써 4개의 대단원으로 구성하였는데 I 단원은 스크래치 프로그래밍 방법에 대하여설명하였고 IV단원은 개발한 프로그램을 발표하는 방법에 대하여설명하였다. 프로그래밍에 관한 대단원은 Ⅱ단원과 Ⅲ단원으로 Ⅱ단원은 게임 등과 같이 문제 해결을 위한 프로그램을 작성하는 주제로 구성하였으며, Ⅲ단원은 프로그램으로 디지털 스토리텔링을 할 수 있도록 구성하였다. 본 교재의 전체 목차는 다음과 같다.

- I. 신나는 프로그래밍을 시작해요
- 1. 프로그램이란 무엇인가요
- 2. 무엇으로 프로그램을 작성하나요
- 3. 스크래치 프로그램은 어떻게 사용하나요
- 4. 프로그램은 어떤 순서로 실행되나요
- Ⅱ. 프로그램은 만능 일꾼
 - 1. 프로그램으로 문제를 해결해요
 - 2. 여기는 외계 행성
 - 3. 우리, 소풍가요
 - 4. 내가 생각하는 동물을 맞혀 보아요
 - 5. 자동차를 운전해요
 - 6. 벌레가 먹이를 찾아요
 - 7. 먹이 빨리 찾기 대회를 열어요
 - 8. 대포를 만들어요
- Ⅲ. 프로그램은 수다쟁이
 - 1. 이야기를 프로그래밍해요
- 2. 우리 오누이가 인사드려요
- 3. 엄마가 떡을 가져오시네요

- 4. 문제를 맞히면 안 잡아먹지
- 5. 호랑이가 오누이를 쫓아와요
- 6. 동아줄을 내려 주세요
- 7. 오누이가 해와 달이 되었어요
- 8. 이야기를 합쳐요

IV. 작품을 발표해요

- 1. 발표할 작품은 어떻게 준비하나요
- 2. 작품 발표회는 어떻게 준비하나요
- 3. 작품 발표는 어떻게 하나요
- 4. 작품 발표가 끝나면 어떻게 하나요

5. 프로그래밍 교재의 평가

5.1 Computational Thinking 요소의 구현

본 교재는 Computational Thinking의 요소를 교육할 수 있도록 개발하였다. 각 요소 교육의 구현 방법은 다 음과 같다.

• 자료 수집, 자료 분석, 자료 표현

자료 수집, 분석 및 표현은 문제를 정확하게 이해하고 문제를 해결하는 데 필요한 자료들을 수집하고 분석하여 표현하는 활동이다. 과제에 따라서는 자료를 수집하지 않는 경우도 발생하며, 경우에 따라서는 자료 외에 유사한 형태의 문제 해결 방법 탐색도 이에 포함시킬 수 있다. 이와 같은 탐색은 과제의 정확한 이해 및 분석이 선행된 후에 이루어질 수 있다. 따라서 본 교재에서는 학생들이 과제를 정확하게 이해하고 분석하는 데 도움을 줄 수 있도록 "다음과 같이 동작하도록 프로그래밍해요" 란을 만들어 제공하였다.

• 문제 분할

문제 분할은 추상화와 더불어 Computational Thinking에서 과제 분석 및 설계 단계에서 중요한 요소이다. 본 교재에서는 주어진 과제를 먼저 줄거리로 분할한 후에 분할된 줄거리를 상세하게 프로그래밍하도록 하였다. 특히 분할된 줄거리를 추가 블록을 사용하여 명시함으로써 문제 분할에 해당하는 사고 활동을 확인할 수 있도록 하였다.

• 추상화

추상화는 불필요한 것을 제거하고 여러 동작을 일 반화된 동작으로 만들어 사용하는 것을 의미한다. 이러한 동작은 여러 동작을 모아서 수행하고자 하 는 기능의 제목을 부여하는 것만으로도 초등학교 수준에서의 추상화 목표를 달성할 수 있다[15]. 따라서 본 교재에서는 주어진 과제를 분할하고 분할 된 문제에 의미 있는 제목을 붙여 추가 블록을 형 성하도록 하였다.

• 알고리즘과 절차

본 교재에서는 과제를 해결하기 위해 필요한 동작들을 표현함으로써 학생들이 프로그래밍을 하기전에 해결해야 하는 동작들에 대하여 안내하여 알고리즘 및 절차를 생각할 수 있도록 하였다.

• 자동화

본 교재에서는 스크래치 프로그램의 스크립트를 보여줌으로써 학생들이 과제를 직접 자동화를 경 험할 수 있도록 하였다.

• 시뮬레이션

본 교재에서는 작성한 프로그램을 실행하는 단계를 명시함으로써 실제로 학생들이 실행할 수 있도록 하였으며, "실행하여 확인해 보세요" 난을 만들어 실행하면서 확인해야 하는 것들을 안내하였다.

• 병렬화

스크래치는 이벤트 블록을 통하여 동시에 수행하는 것이 가능하므로 기본적으로 병렬화의 구현이 기본적으로 가능하다. 본 교재에서는 이에 대하여 학생들이 이해할 수 있도록 설명하였다.

5.2 학생에의 적합도

앞에서 언급한 바와 같이 Computational Thinking의 각 요소를 교육하기 위하여 여러 방안을 구안하여 본 교재를 개발하였다. 그러나 이러한 방안들도 학생들에게 적합하지 않다면 목적을 달성하기 힘들 것이다. 따라서 본 교재가 학생들에게 적합한 내용과 구성을 가지고 있는지 조사한 결과는 다음과 같다.

(1) 흥미도

본 교재의 학생들에 대한 적합도를 조사하기 위하여 경기도 ○○초등학교의 동아리의 학생 14명에게 시범수업을 실시하였다. 시범 수업의 내용은 Ⅱ부 5장인 "자동차를 운전해요" 단원이다. 시범 수업을 실시하고 난후에 학생들에게 설문 조사한 결과는 다음과 같다.

수업을 실시한 후에 학생들의 흥미도에 대하여 조사한 결과는 <Table 2>와 같다.

<Table 2> Survey on Interesting

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	0	0	0	7	7
2	0	0	1	4	9
3	0	0	0	5	9
4	0	0	2	5	7
(5)	0	0	0	4	10

- ① "Did you actively participate in class?"
- 2 "Were you interested in this subject?"
- ③ "Do you want to continue to learn this subject?"
- ④ "Were you increase the interest in the programming?"
- ⑤ "Do you want to recommend software class to your friends?"

<Table 2>의 결과와 같이 학생들은 전반적으로 수 업에 대하여 상당한 흥미를 나타내었다. 특히, 많은 학 생들이 친구들에게도 추천하고 싶어 하는 것으로 나타 나 흥미도면에서는 좋은 평가를 받았다.

5.3 교재의 구성

동일한 시범 수업 후에 교재의 구성이 학생들에게 적절한지를 조사한 결과는 <Table 3>과 같다.

< Table 3> Survey on the Structure of the Material

ς	Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
	1	0	0	1	4	9
	2	0	0	2	3	9
	(3)	0	0	1	5	8

- ① "Did you read this material comfortably?"
- 2 "Did you know what you learn in this material?"
- ③ "Could you easily learn the contents in this material?"

교재가 학생들에게 적합한지를 묻는 질문에서도 학생들은 충분히 이해할 수 있다고 대답해서 교재의 구성도 초등학생들의 수준에 타당한 것으로 나타났다.

6. 결론 및 추후 과제

2019학년도부터 초등학교 5~6학년에서 소프트웨어 교육이 실시될 예정이다. 소프트웨어 교육에서의 핵심 내용은 프로그래밍 교육이라고 할 수 있다. 여기에서 언급하는 프로그래밍 교육은 개발자 양성 교육이 아니 라 학생들에게 정보 사회에서 필요한 Computational Thinking을 배양하기 위한 교육이다. Computational Thinking은 자동화를 목적으로 문제를 해결하기 위한 방법이다. 따라서 문제를 이해하고, 문제 분할 및 추상 화 과정을 통하여 문제 해결 방법을 사고하며 이를 프 로그램으로 작성하는 과정이라고 할 수 있다. 그러나 기존의 초등학생용 프로그래밍 교재들은 따라하기 위주 의 매뉴얼 형태로 만들어진 것이 대부분이기 때문에 소 프트웨어 교육의 목적을 달성하기 어려운 문제점이 있 었다. 따라서 본 연구에서는 초등학교의 동아리 활동 시간에 활용할 수 있는 프로그래밍 교재를 개발하였다. 즉 문제를 이해하고, 문제 분할 및 추상화 과정을 명시 적으로 표현하게 하며, 인지 도제 방법을 구현하여 학 생들이 Computational Thinking을 체계적으로 습득할 수 있도록 하였다. 또한 시범 수업을 통하여 초등학생 들에게 적합하며 수준에 맞는 내용임을 입증하였다.

그러나 본 교재의 개발 목적인 학생들의 Computational Thinking 능력 개발을 측정하기 어려웠다. Computational Thinking 능력 개발에 유효함을 입증하기 위해서는 보다 많은 학생들에게 보다 장기간의 교육을 통하여 입증할 수 있기 때문이다. 다만 기존 교재들의 문제점을 해결할 수 있는 방안을 제시하고 있다는 점에 의의를 둘수 있을 것이다. 따라서 본 교재 개발 시에 사용되었던 방법들에 대한 효과성은 계속 연구되어야 할 것이다.

이와 함께 학생들의 다양한 수준과 관심에 부응할 수 있도록 보다 다양한 프로그래밍 교육 교재의 개발이 시 급하다.

참고문헌

- [1] Bae, HakJin, Lee, EunKyoung, & Lee, YoungJun (2009). A Problem Based Teaching and Learning Model for Scratch Programming Education. The Journal of Korean Association of Computer Education, 12(3), 11–22.
- [2] Baek, Seon-Ryeon, Song, Jeong-Beom, Park, Jung Ho, & Lee, Tae-Wuk (2008). Development and Application of Algorithm Teaching Materials Centered on Plays for Problem-solving Abilities of Elementary Students. The Journal of Korean Association of Computer Education, 11(1), 85-95.
- [3] Chae, Kyeong Jeon, Kim, Jong Han, & Kim, Taeyoung (2012). The Instructional Design of Programming based on Discovery Learning Model for Elementary School Students. *Proceedings of the* KACE, 16(1), 52–56.
- [4] Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics. In L. B. Resnick (Ed.) Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser. Hillsdale, NJ: Lawrence Erlbaum Associates.
- [5] Dennen, V. P., & Burner, K. J. (2008). The Cognitive Apprenticeship Model in Educational Practice. In J. M. Spector, M. D. Merrill, J. V. Merrienboer, & M. P. Driscoll (eds.) Handbook of Research on Educational Communication and Technology. 3rd edition. New York; Lawrence Erbaum Associates.
- [6] Guzdial, M. (2004). Programming Environments for Novices. Computer Education Research, 127–154.
- [7] Jeong, Inkee (2014). Comparison and Analysis of Visual Programming Tools for Coding Education for K-12. The Korean Association of Information Education Research Journal, 5(3), 63-70.
- [8] Jeoung, Mi Yeoun, Lee, Eun Kyoung, & Lee, Young Jun (2008). The Effects of Algorithm Learning with Squeak Etoys on Middle School Students' Problem Solving Ability. The Journal of Korean Institute of

- Industrial Education, 33(2), 170-191.
- [9] Jo, M. H., & Lee, Y. H. (1994). The Direction of Instructional Design Adopting Cognitive Apprenticeship. Journal of Educational Technology, 9(1), 147–162.
- [10] Kim, Chul (2012). An Analysis of Domestic Research Trend and Educational Effects in Relation to Robot Education. Journal of The Korean Association of Information Education, 16(2), 233–243.
- [11] Kim, Kyung-Hoon, Kang, Oh-Han, Kim, Yung-Sik, Kim, Yoon Young, Soon Seo In, Ahn, Seong Jin, Jung, Soon Young, & Choe, Hyun Jong (2012). A Study on the Direction of Informatics Education Strategies Based on the Creative Problem Solving to Improve Core Competencies. KICE Report RRC 2012-7.
- [12] Kim, Yoon Young, & Kim, Yungsik (2013). Effect of Learning a Divide-and-conquer Algorithm on Creative Problem Solving. *The Journal of Korean Association of Computer Education*, 16(2), 9-18.
- [13] Lee, EunKyoung (2009). A Task Centered Scratch Programming Learning Program for Enhancing Learners' Problem Solving Abilities. The Journal of Korean Association of Computer Education, 12(6), 1–9.
- [14] Lee, Joo Heon (1993). Applications of Software Engineering Principles. BobYoung Sa.
- [15] Lee, Young Jun, Paik, Seoung Hey, Hong, Shin Jae, Yu, HeonChang, Jeong, Inkee, An, SangJin, Choi, JeongWon, & Jeon, SeongKyun (2014). Research for Introducing Computational Thinking into Primary and Secondary Education. Korea Foundation for the Advancement of Science and Creativity.
- [16] Moon, Wae-Shik (2006). Analysis of Error Types occurring on Elementary School Student's Programming Learning. Journal of the Korea society of computer and information, 11(2), 319–327.
- [17] Novick, Laura R., & Bassok, Miriam (2005).

- Thinking and Reasoning Chapter 14 Problem Solving. Cambridge University Press.
- [18] Papert, S. (1980). MINDSTORMS: Children, Computers, and Powerful Ideas. Basic Books Inc.
- [19] Park, YoungSun, An, SangJin, & Lee, YoungJun (2013). A Differentiated Instruction Model for Elementary School Student on Programming Education Using Scratch. *Proceedings of the* KACE, 17(2), 167–170.
- [20] Seol, Moon-gu, & Son, Chang-ik (2013). A Study on Development of Teaching Materials for App Inventor Programming Using the Waterfall Model. Journal of The Korean Association of Information Education, 17(4), 409-419.
- [21] Song, Jeong-Beom, Cho, Soeng-Hwan, & Lee, Tae-Wuk (2008). The Effect of Learning Scratch Programming on Students' Motivation and Problem Solving Ability. Journal of The Korean Association of Information Education, 12(3), 323-332.
- [22] Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*(3), 33–35.
- [23] Yang, Changmo (2014). Meta-Analysis on the Effects of Programming Education using Educational Programming Languages. Journal of the Korean Association of Information Education. 18(2), 317-324.
- [24] Yeon, Hyejin, Jo, Miheon (2014). Development of a Robot Programming Instructional Model based on Cognitive Apprenticeship for the Enhancement of Metacognition. Journal of The Korean Association of Information Education, 18(2), 225–234.
- [25] Yu, Jeong Su, Lee, Min Hui (2009). Effects of a Programming Class Using Dolittle on Enhancing Creativity, Problem Solving Ability. and Interest in Programming. *Journal of The Korean Association of Information Education*, 13(4), 443–450.

저자소개



정 인 기

- 1988 고려대학교 전산과학과(이 학사)
- 1990 고려대학교 대학원 수학과 전 산학전공(이학석사)
- 1996 고려대학교 대학원 전산과학 과 전산학전공(이학박사)
- 1997~현재 춘천교육대학교 컴퓨 터교육과 교수
- 관심분야: 컴퓨터과학교육, 프로 그래밍 교육

e-mail: inkey@cnue.ac.kr