

Development of Python Education Program with Computational Thinking

Min-Kyung Lee*

*Assistant Professor, Dept. of College of General Education, Silla University, Busan, Korea

[Abstract]

In this paper, we propose a python education program that applies computational thinking for non-majors and programming beginners. In this study, we focus on the basics of program logic, breaking away from the difficult grammar and memorization-oriented programming education. And by applying the problem-solving procedure of computational thinking, we propose an educational program that allows non-majors and programming beginners to learn programming easily. In this paper, an 8-week educational program was applied to middle school students with little text coding experience. and through a post-satisfaction survey, it was found that their confidence in programming increased, and they were able to apply computational thinking could be applied to life and other subjects. Although the importance of programming education is being emphasized, it is expected that it will be used as a useful educational program when composing program education for non-majors and beginners in programming for learners who still find it difficult to learn programming.

▶ **Key words:** Computational Thinking, Software Education, Python Programming, Informatics Education, Text-based programming language

[요 약]

본 논문에서는 비전공자 및 프로그래밍 초보자들을 위한 컴퓨팅 사고력을 적용한 파이썬 교육 프로그램을 제안한다. 본 논문은 어려운 문법 위주, 암기 위주의 프로그래밍 교육에서 벗어나 프로그램 논리의 기본기에 충실하고 컴퓨팅 사고력의 문제 해결 절차를 적용하여 비전공자 및 프로그래밍 초보자들도 쉽게 프로그래밍 학습을 할 수 있는 교육 프로그램을 제안하며, 텍스트 코딩 경험이 적은 중학생들을 대상으로 8주간 교육 프로그램을 적용하였고 사후 만족도 설문 조사를 통해 프로그래밍에 대한 자신감이 높아졌으며 일상생활이나 다른 교과목에도 컴퓨팅 사고력을 적용할 수 있게 되었다는 점을 알 수 있었다. 프로그래밍 교육의 중요성이 강조되고 있지만 여전히 프로그래밍 학습에 어려움을 느끼는 학습자들을 위해 향후 비전공자 및 프로그래밍 초보자들을 위한 프로그램 교육을 구성할 때 유용한 교육 프로그램으로 활용될 것으로 기대한다.

▶ **주제어:** 컴퓨팅사고력, 소프트웨어 교육, 파이썬 프로그래밍, 정보교과, 텍스트 기반 프로그래밍 언어

• First Author: Min-Kyung Lee, Corresponding Author: Min-Kyung Lee
*Min-Kyung Lee (mklee@silla.ac.kr), Dept. of College of General Education, Silla University
• Received: 2022. 10. 26, Revised: 2022. 11. 11, Accepted: 2022. 11. 16.

I. Introduction

4차 산업혁명으로 인해 시작된 초 지능, 초 연결 사회는 디지털 기기를 중심으로 빠르게 진행되고 있으며, 우리의 미래 모습을 바꿔 놓을 핵심 기술들 중에서 소프트웨어가 중요한 부분을 차지하게 되었는데, 디지털 기기를 중심으로 빠르게 진행되고 있는 사회, 경제 및 직업의 변화 속에서 소프트웨어는 핵심 기술로 중요한 부분을 차지하고 있기 때문에 학교 교육에서도 디지털 시대에 역량 있는 인재 육성을 위해서 소프트웨어를 활용하고 개발하는 방법을 배우는 소프트웨어 교육은 선택이 아니라 필수가 되었다[1][2].

소프트웨어를 중심으로 빠르게 변화하는 사회에 적응하기 위해 컴퓨팅 사고력, 창의융합적 사고력 등의 새로운 역량을 갖춘 인재의 배출을 위해 학생들은 자기 주도적인 학습 태도와 더불어 창의적인 역량도 필요하게 되었으며 초등학교, 중학교, 고등학교 정규교육 과정에서도 이를 위한 소프트웨어 교육을 강화하고 있다[3].

우리나라 소프트웨어 교육과정은 2014년에 알고리즘과 프로그래밍 교육과정으로 확대되었고, 2015년 개정 교육과정에서는 2018년도 중학교 정보 교과를 필수로, 고등학교에서는 선택과목으로 강화하였으며, 2019년부터는 초등학교에서도 필수로 소프트웨어 교육과정을 적용하여 소프트웨어 교육의 강화와 창의융합적 사고력을 갖춘 인재 양성에 중점을 두고 있다. 대학의 경우도 2019년부터 소프트웨어 중점대학 지원 사업을 통해 소프트웨어 교육을 의무화하고 있다[4].

국내의 소프트웨어 교육 관련 연구의 동향을 보면 컴퓨팅 사고력 교육과 소프트웨어 교육이 중요한 패러다임으로 자리 잡고 있으며, 소프트웨어 교육은 연령과 교육 환경에 따라 효율적인 프로그래밍 언어를 활용하여 다양한 교육 방법이 시도되고 있다. 그리고 교육에 사용된 프로그래밍 언어는 2017년 이후 파이썬이 비전공자 대상 컴퓨팅 사고 교육에 자주 활용되고 있는 것으로 나타난다[5].

컴퓨팅 사고력은 종합적인 사고 과정이며, 프로그래밍 교육은 잠재된 사고력을 끄집어내어 구체적으로 표현하도록 훈련하는 방법이다. 이처럼 프로그래밍 교육은 중요하게 강조되고 있지만 실제로 교육 현장에서 학생들은 프로그래밍 교육을 어렵게 느끼고 있는 현실이며, 교육현장에서 프로그래밍 수업을 어떻게 설계하고 적용할 수 있을지에 대한 다양한 해결 방안들이 제안되고 있다[6][7][8].

변화하는 시대의 핵심 역량이라 할 수 있는 컴퓨팅 사고력의 신장을 위한 프로그래밍 교육은 컴퓨팅 사고력을 적용하여 문제해결능력을 향상시키는 것이다.

그러나 여전히 교육현장에서 학습자들은 프로그래밍 교육에 어려움을 느끼고 있는 현실이므로 비전공자 또는 프로그래밍 초보자인 학습자들이 쉽게 접할 수 있는 프로그래밍 언어를 사용하여 프로그래밍 초보자들도 쉽게 학습할 수 있도록 교육 프로그램을 구성하고 프로그래밍에 흥미를 잃지 않도록 할 필요가 있다.

파이썬은 1991년 귀도 반 로섬이 발표한 고급 프로그래밍 언어로 데이터 과학, AI 및 범용 소프트웨어 개발에 이르기까지 다양한 분야에서 중요한 프로그래밍 언어로 자리매김하고 있으며, 쉽고 직관적인 문법으로 쉽게 배울 수 있어 프로그래밍 초보자를 위한 교육뿐 아니라 다양한 분야에서 많이 활용되고 있다[9][10][11].

이에 본 연구에서는 비전공자 및 프로그래밍 초보자인 학습자들에게 적용할 수 있는 컴퓨팅 사고력을 적용한 파이썬 프로그래밍 교육 프로그램을 제안한다.

교육과정은 문법 위주의 소프트웨어 교육보다는 컴퓨팅 사고력을 적용하여 실생활이나 전공과 무관하게 모든 영역에서 창의적으로 문제를 해결할 수 있는 능력을 기를 수 있도록 문제 해결에 컴퓨팅 사고력을 적용할 수 있도록 설계하였으며 학습자들의 흥미를 유발하고 직관적으로 프로그래밍 결과를 이해할 수 있도록 터틀 모듈을 활용하여 설계하였다.

본 논문의 구성은 2장에서 본 논문과 관련된 연구에 관해 기술하고, 3장에서는 본 연구에서 제안하는 컴퓨팅 사고력을 적용한 파이썬 교육 프로그램의 개발 과정과 적용과정을 기술한다. 4장에서는 결론에 관해 기술한다.

II. Preliminaries

1. Computational thinking

컴퓨팅 사고력(Computational Thinking)은 컴퓨터를 중심으로 빠르게 진행되고 있는 4차 산업혁명 시대와 21세기를 살아가는 모든 사람이 갖추어야 할 기본적인 핵심 역량으로 컴퓨터 관련 분야에만 국한되지 않고 예술, 인문, 사회 등 다양한 분야에서 필요한 중요한 역량으로 확산되고 있다[12][13][14].

컴퓨팅 사고력은 Seymour Papert 교수가 기하학적 아이디어 생성을 위한 방법론으로 처음 제시하였으나, 2006년 Jeannette M. Wing 교수가 ACM에 발표한 논문을 통해서 전 세계적으로 알려지기 시작했으며, 컴퓨터 과학의 기본 개념을 바탕으로 창의적인 문제 해결을 위해 21세기의 모든 사람들이 갖추어야 할 사고 능력이라고 규정하였다[1][15].

컴퓨팅 사고력은 컴퓨터 과학의 원리와 개념을 바탕으로 문제를 해결하는 연속적이고 체계적인 사고의 과정이며, 학자와 연구기관마다 개념과 속성이 서로 다르게 정의되고 분류되어 있다. 컴퓨팅 사고력의 구성요소에 대한 정의는 다양한데 대표적인 내용을 정리하면 표 1과 같다 [16][17][18][19].

2. Computational thinking and software education

소프트웨어 교육은 디지털 시대 학습자들이 갖추어야 할 핵심 역량으로 논의되는 컴퓨팅 사고력을 향상시키는 효과적인 도구로 여겨지고 있으며, 소프트웨어 교육이 컴퓨팅 사고력 향상에 효과가 있다는 사실을 다양한 연구에서 증명하고 있다[20][21][22][23].

소프트웨어 중심의 멀티미디어 사회에서 우리 일상의 많은 부분이 소프트웨어 중심으로 돌아가고 있으며, 프로그래밍 과정에서 경험하게 되는 문제 분해, 추상화, 알고리즘 작성 등의 역량은 컴퓨팅 사고력의 중요 구성요소들이므로 결과적으로 소프트웨어 교육이 학습자들의 컴퓨팅 사고력 향상을 위한 중요한 방법이라고 할 수 있다[24].

우리나라의 소프트웨어 교육 활성화 정책으로 컴퓨팅 사고력과 소프트웨어 교육은 초등학교, 중학교, 고등교육에서 필수 패러다임으로 자리 잡고 학습자의 교육과정에 알맞은 프로그래밍 언어를 활용한 소프트웨어 교육을 통해 창의적인 문제해결 능력 향상을 위한 효율적인 소프트웨어 교육과 관련된 다양한 연구들이 진행되고 있다[5].

3. Text programming language python

프로그래밍 초보자들의 소프트웨어 교육을 위해 텍스트 코딩보다 블록 코딩을 더 선호하는 이유는 배우기 쉽다는 장점 때문이지만 기능에 한계가 있으며, 텍스트 프로그래밍 언어는 블록 프로그래밍 언어와 비교해서 다른 텍스트 프로그래밍 언어로의 전이 가능성도 높아서 컴퓨팅 사고력 향상에 도움이 된다는 연구 결과들이 많다[25].

파이썬(Python)은 1991년 네덜란드의 귀도 반 로섬(Cuido van Rossum)이 발표한 5세대 프로그래밍 언어이며 간단하고 이해하기 쉬운 문법과 플랫폼에 독립적이라서 여러 운영체제에서 사용이 가능하고 다양한 자료형의 제공과 라이브러리 제공으로 교육용 프로그래밍 언어에서부터 인공지능 분야에 이르기까지 다양하게 사용되고 있으며, 터틀 모듈(Turtle Module)을 활용하여 실행 결과를 시각적으로 바로 확인하는 것도 가능해서 프로그래밍 초보자들이 텍스트 프로그래밍에 대한 거부감을 가지지 않고 쉽고 재미있게 소프트웨어 교육을 할 수 있다[26].

Table 1. Components of computational thinking

Researcher	Components
Wing (2006)	<ul style="list-style-type: none"> • Problem Expression • Redefining the Problem • Recursive Thinking • Decomposition • Abstract • Heuristic Inference • Generation • Modularity • Forward Thinking • Efficiency of Time and Space
ISTE&CSTA (2011)	<ul style="list-style-type: none"> • Data Collection • Data Analysis • Data Representation • Problem Decomposition • Abstraction • Algorithm&Procedure • Automation • Simulation • Parallelization
KERIS (2015)	<ul style="list-style-type: none"> • Data Collection • Data Analysis • Structuring • Abstract <ul style="list-style-type: none"> - Decomposition - Modelling - Algorithm • Automation - Coding - Simulation • Generalization
Google (2016)	<ul style="list-style-type: none"> • Decomposition • Pattern Recognition • Abstraction • Algorithm • Design

4. Software education for non-majors

프로그래밍 학습은 오랫동안 어려운 과목으로 인식되어 왔는데 기존의 학습 방법이 프로그래밍 사용법이나 문법 위주의 실행, 암기 위주여서 학습자에게 부담감을 주었기 때문이다. 대학의 교양과정으로 프로그래밍 교육을 하는 대학생들 중 본인의 전공이 인문계나 예체능인데 프로그래밍을 왜 배워야 하는지 의구심을 가지며 프로그래밍 교육의 필요성을 느끼지 않거나 심지어 기피하는 경우도 많아서 이 부분을 해소하기 위한 많은 연구들이 있다[27].

컴퓨팅 사고력 함양을 위한 프로그래밍 교육은 크게 블록 기반 언어 교육과 텍스트 기반 언어 교육으로 나뉘는데 비전공자들을 위한 텍스트 기반 언어는 문법이 쉬운 파이썬을 주로 사용하고 있다[16].

전공자는 물론 비전공자도 4차 산업혁명의 핵심 역량인 창의·융합 사고력을 키우기 위한 노력은 많이 하고 있지만 비전공자들이나 프로그래밍 초보자들은 여전히 학습에 어려움을 느끼고 있으므로 본 연구에서는 가능한 복잡한

문법을 배제하고 기본 논리를 위주로 컴퓨팅 사고력의 절차에 따라 프로그래밍을 학습할 수 있도록 교육 프로그램을 구성하였다.

III. The Proposed Scheme

1. Education program proposal

본 논문에서 제안하는 교육 프로그램은 이전에 컴퓨팅 사고력을 적용한 수업에 대한 경험이 없고 프로그래밍 교육 또한 경험이 없거나 블록 코딩의 경험만 있는 프로그래밍 초보자들을 대상으로 하는 프로그램으로 설계하였다.

교육 프로그램은 파이썬 프로그래밍 언어를 사용하여 설계되었으며 수업 내용의 범위는 컴퓨팅 사고력의 개념과 적용 방법 그리고 파이썬 프로그래밍 언어의 기본 개념부터 프로그램 논리인 순서, 선택, 반복 논리까지이다.

수업의 범위를 프로그램 논리인 순서, 선택, 반복까지로 한정하는 이유는 텍스트 코딩을 처음 배우는 초보자들이 프로그래밍 학습에 어려움을 느껴 중도 포기함이 없도록 단기간 내에 교육과정을 이수하고 성취감을 얻을 수 있도록 하기 위함이다.

또한 지속적인 관심과 흥미를 통해 이후 일반 실생활뿐 아니라 다른 교과목에도 컴퓨팅 사고력을 적용하여 창의적인 문제 해결을 할 수 있는 역량을 기르고 더불어 소프트웨어적 역량을 더욱 향상시키기 위한 발판으로 삼기 위해 프로그램 과정을 최대한 단순화하였다.

2. Application

본 논문의 교육 프로그램은 부산 시내 소재 중학생 27명을 대상으로 8주간 교육 프로그램을 적용하였다.

학생들은 1학년 15명, 2학년 4명, 3학년 8명 총 27명이며 이전에 컴퓨팅 사고력을 적용한 수업을 들어본 경험이 있는 학생은 3명이었는데 아두이노 등의 피지컬 컴퓨팅 수업을 통해 컴퓨팅 사고력 수업을 경험해 본 적이 있다고 답했다.

그림 1은 학생들의 프로그래밍 교육 경험에 대한 설문 결과이다. 27명 중 1명을 제외한 모든 학생은 프로그래밍 수업을 한 적이 있다고 답했는데 프로그래밍의 종류는 응답자 26명 중 피지컬 컴퓨팅(9명) 34%, 블록 코딩 프로그래밍(14명) 53%, 텍스트 코딩(3명) 12% 순으로 나타났다.

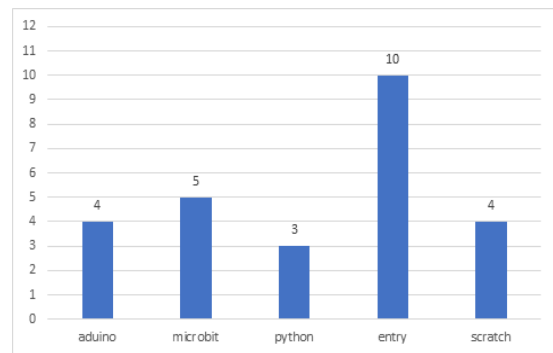


Fig. 1. Programming Learning Experience

3. Computational thinking process

컴퓨팅 사고력의 구성요소에 대한 정의는 다양하지만 본 연구에서는 KERIS (2015)의 구성요소를 단순화 시켜서 그림 2와 같이 문제분석(Analysis), 알고리즘(Algorithm), 자동화(Automation)의 3단계로 구성하였으며, 각 단계는 절차에 따라 순서대로 진행된다.

각 단계에서 수행할 내용은 다음과 같다.

첫째, Analysis 단계는 문제를 분해하는 단계로 자료수집, 자료 분석을 위한 단계이다. 문제 해결에 필요한 변수, 수식 또는 공식을 찾고 문제 해결을 위해 적용되는 프로그램 논리(순서, 선택, 반복)를 찾는 단계이다.

둘째, Algorithm 단계는 추상화 단계의 구성요소 분해, 모델링, 알고리즘 3단계를 알고리즘 단일 단계로 재구성하여 분해된 문제를 프로그램 논리(순서, 선택, 반복)를 적용하여 순서도나 자연어 방식으로 문제 해결의 절차를 표현하는 단계이다.

셋째, Automation 단계는 코딩, 시뮬레이션의 구현 단계로 알고리즘 단계에서 작성된 순서도나 자연어를 절차에 따라 파이썬 프로그램으로 작성하고 시뮬레이션 하는 단계이다.

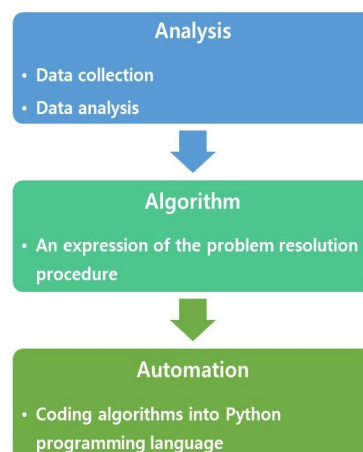


Fig. 2. Computational thinking process

4. Turtle module

파이썬은 다양한 자료형과 다양한 라이브러리를 제공하는 비교적 배우기 쉬운 언어이며 그래픽 기능도 단순해서 초보자들도 쉽게 사용할 수 있는 프로그래밍 언어이다.

터틀 모듈(Turtle Module)은 프로그래밍의 결과를 그래픽으로 즉시 확인할 수 있도록 하는데 프로그래밍 초보자들의 경우 텍스트 코딩으로만 결과를 확인하는 것보다는 본인이 작성한 프로그램에 대한 결과를 시각적으로 바로 확인할 수 있도록 하는 것이 프로그래밍에 대한 흥미도를 높이고 자신감도 향상시킬 수 있다.

그림 3은 터틀 모듈을 이용해서 반복문으로 사각형을 그리는 코드이며 그림 4는 프로그램 실행 결과이다.

```

1 import turtle
2 t=turtle.Turtle()
3
4 i=0
5 while i<4:
6     t.forward(100)
7     t.right(90)
8     i=i+1
9
    
```

Fig. 3. Coding for drawing squares

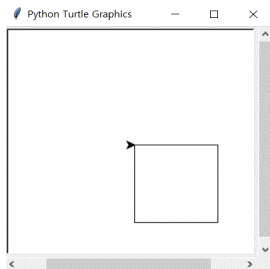


Fig. 4. Turtle execution result

위 그림 3, 그림 4에서 보는 것처럼 터틀 모듈(Turtle Module)을 이용하면 프로그램의 실행 결과를 직접 그래픽으로 확인할 수 있어서 프로그래밍 원리를 보다 직관적으로 확인할 수 있다.

5. Education program design

미래사회의 핵심 역량인 창의·융합적 인재 양성을 위한 컴퓨팅 사고력을 적용한 파이썬 교육 프로그램은 학습자들이 재미있고 쉽게 프로그래밍의 기본 논리를 이해할 수 있도록 설계하였다.

본 교육과정은 프로그래밍 교육을 해 본 적이 없거나 텍스트 코딩의 경험이 없는 프로그래밍 초보 학습자들이 프로그램이 어렵다는 선입견을 가지지 않고 프로그래밍 교육에 지속적인 흥미를 가질 수 있도록 가능한 쉬운 예제들을 중심으로 개발하였고 교육 프로그램 과정은 표 2와 같다.

Table 2. Python programming curriculum with computational thinking

Week	Topic	Learning contents
1	Computational thinking	Computational overview Marshmallow Challenge with Computational Thinking
2	Python language overview	variable, data type print() function input() function
3	Turtle Library Standard I/O	Drawing a figure conversation program arithmetic program
4	Conditional structure	login program arithmetic program
5	Conditional structure	even & odd number discrimination program drawing polygons using turtle.
6	Iteration structure	a program that prints a number from 1 to the number entered multiplication table program drawing polygons with a turtle
7	Conditional structure & Iteration structure	number guessing program calculator program
8	Conditional structure & Iteration structure	vending machine calculator program turtle racing program

표 2의 교육 프로그램은 8주간의 과정으로 파이썬 프로그래밍의 기본 개념과 프로그래밍 언어의 기본 논리인 순서, 선택, 반복 논리를 이해하고 활용할 수 있도록 구성되었으며 3주, 5주, 6주, 8주에는 터틀 모듈을 이용하여 선택 논리와 반복 논리를 보다 직관적으로 이해할 수 있도록 구성하였다. 3주, 4주, 7주에 중복되어 실시한 연산 프로그램은 연산 문제를 해결 하는 과정에 순서, 선택, 반복 논리를 모두 적용하여 문제를 해결하는 방법은 한 가지 논리만을 적용해서 해결하는 것이 아니고 여러 가지 방법으로 해결할 수 있음을 보이기 위함이다. 그리고 모든 문제를 해결하는 과정에서 컴퓨팅 사고력의 절차를 적용하기 위해서 학습지를 이용하도록 하였다.

6. Differences

본 논문에서 제시한 교육 프로그램의 차이점은 문제를 위해 컴퓨팅 사고력의 단계를 수행하는 학습지를 사용하였다는 점이다. 6.1과 6.2는 마시멜로 챌린지와 연산 프로그램을 작성하기 위해 학습지를 이용하여 컴퓨팅 사고력의 단계를 수행한 예시이다.

6.1 Marshmallow challenge

1주 차에 실시한 마시멜로 챌린지는 컴퓨팅 사고력을 프로그래밍 문제에 바로 적용하지 않고 팀 활동을 통해 먼저 체험하는 과정인데 이러한 활동은 학습자들이 컴퓨팅 사고력을 협동을 통해 놀이처럼 체험하게 하여 학습자의 흥미를 유발하고 다음 과정인 프로그래밍 문제에 컴퓨팅 사고력을 적용하는 것을 용이하게 한다.

마시멜로 챌린지 문제를 컴퓨팅 사고력을 적용하여 해결하는 과정은 다음과 같다.

(1) Analysis

마시멜로 챌린지의 문제 해결을 위해 스파게티로 쌓아 올린 탑이 튼튼하고 높게 일정 시간 유지되도록 구조물의 바닥 설계와 모양을 어떻게 만들어야 할지에 대한 내용들을 컴퓨터나 스마트폰을 이용하여 자료를 조사하고 타당한 이유를 찾도록 한다.

(2) Algorithm

Analysis 과정에서 적절한 바닥 모형과 탑의 모양을 결정했다면 주어진 재료들을 가지고 탑을 만들어갈 순서를 자연어나 순서도를 이용해서 표현한다. 본 연구에서 알고리즘 단계는 추상화 단계를 간소화해서 절차만을 표시하는 단계이지만 마시멜로 챌린지 활동에서는 모델링 과정을 포함하여 학습자들이 조사한 자료를 이용해서 만들어질 탑을 간략하게 모델링 하는 과정을 포함하여 활동을 하였다.

(3) Automation

프로그래밍에서는 알고리즘을 실제로 코드로 작성하는 단계이지만 마시멜로 챌린지 문제에서는 작성한 알고리즘에 따라 재료들을 이용하여 실제로 스파게티로 탑을 만드는 과정이다.

그림 5는 실제 탑을 제작하기 전 학습지를 통해 컴퓨팅 사고력의 절차를 수행한 예제이고, 마지막 Automation 단계에서 팀별로 직접 주어진 재료를 이용해서 탑을 만들었는데 학습자들이 이 과정을 통해 컴퓨팅 사고력의 절차를 자연스럽게 학습하게 된다.

6.2 Arithmetic programming

그림 6은 두 개의 숫자를 더한 후 합계를 출력하는 아주 단순한 문제이다. 그러나 실제로 학습자들은 이 문제를 해결하기 위한 절차와 방법을 쉽게 알아내지 못하는 경우가 많다. 그러나 그림 6에서처럼 컴퓨팅 사고력의 절차를 단계적으로 수행하면서 변수와 수식을 찾고, 알고리즘을 작성한 다음 코딩을 수행하면서 문제를 논리적으로 해결해 나가는 능력을 키울 수 있다.

문제문제 (자료수집/ 분석)	문제 문제 (num1) 문제 (num2) 문제 (sum) 문제 sum = num1 + num2
추상화 (알고리즘)	문제 문제 문제 문제
자동화 (코딩/ 시뮬레이션)	<pre> num1=int(input{"숫자 입력:"}) num2=int(input{"숫자 입력:"}) sum=num1+num2 print("num1+num2=",sum) </pre>

Fig. 6. Arithmetic programming Computational Thinking Activity Sheet

문제문제 (자료수집/ 분석)	문제 문제 문제
추상화 (알고리즘)	문제 문제 문제
자동화 (코딩/ 시뮬레이션)	문제 문제 문제

Fig. 5. Marshmallow Challenge Computational Thinking Activity Sheet

7. Curriculum satisfaction survey

학습자들을 대상으로 만족도 조사를 한 결과는 다음과 같다.

그림 7은 교육 과정은 만족합니까? 라는 질문에 대한 응답이며 81%가 매우 그렇다고 답하였다.

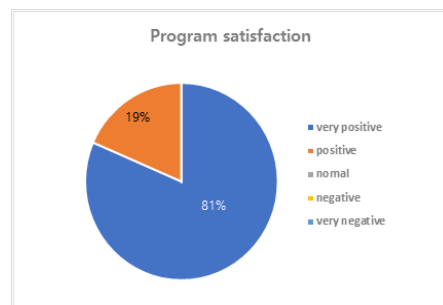


Fig. 7. Program satisfaction

그림 8은 교육 과정 이전과 비교했을 때 프로그래밍에 대한 자신감이 생겼는가? 라는 질문에 대한 응답이며 74%가 매우 그렇다고 답하였다.

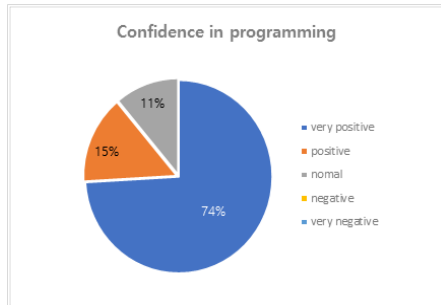


Fig. 8. Confidence in programming

그림 9는 컴퓨팅 사고력을 실생활이나 수학, 과학 등의 다른 교과목에 적용할 수 있는가? 라는 질문에 대한 응답이며 52%가 매우 그렇다고 답하였다.

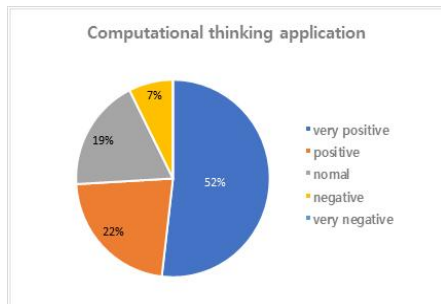


Fig. 9. Computational thinking application

그림 10은 다른 친구들에게도 교육과정을 추천하겠는가? 라는 질문에 대한 응답이며 85%가 매우 그렇다고 응답하였다.

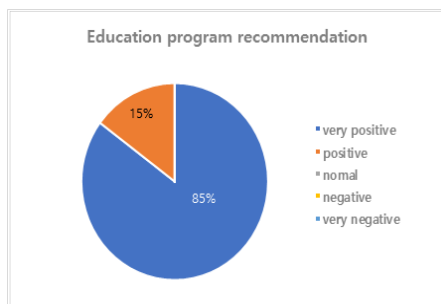


Fig. 10. Education program recommendation

기타 교육 과정에서 가장 좋았던 점을 적어주세요. 라는 주관식 질문에는 “어떻게 문제를 해결할 수 있는지 알 수 있었다”, “프로그래밍의 기초 과정을 꼼꼼하게 다지는 좋

은 수업이었다.”, “팀원들과 협력하는 능력을 길렀다.” 등의 의견이 있었다.

IV. Conclusions

소프트웨어 교육에서 학습자의 교육과정에 알맞은 프로그래밍 언어를 활용하는 것은 중요한 부분이다. 비전공자 들이나 프로그래밍 초보자들은 비교적 코딩이 쉬운 블록 코딩을 이용한 교육을 많이 하게 되지만 텍스트 프로그래밍 언어는 블록 프로그래밍 언어와 비교해서 다른 텍스트 프로그래밍 언어로의 전이 가능성도 높아서 컴퓨팅 사고력 향상에 도움이 된다는 연구 결과들이 많다[24].

본 논문에서는 비전공자 및 프로그래밍 초보자들이 컴퓨팅 사고력을 적용하여 프로그래밍 언어의 기본 논리인 순서, 선택, 반복 논리를 쉽게 배울 수 있도록 컴퓨팅 사고력을 적용한 파이썬 교육 프로그램을 제안하였다.

중학생을 대상으로 8주간의 교육 과정을 진행한 후 만족도 조사를 하였으며 문제를 해결하는 모든 과정에 컴퓨팅 사고력의 절차를 수행하기 위한 학습지를 이용하였다.

그리고 파이썬 터틀 모듈(Turtle Module)을 이용하여 프로그램 실행 결과를 직관적으로 확인할 수 있도록 하여 학습자들의 흥미와 관심을 유도하였다.

사후 만족도 조사에서 학습자들은 교육과정에 만족하고 이전과 비교했을 때 프로그래밍에 대한 자신감도 높아졌지만 다른 과목이나 실생활에 컴퓨팅 사고력을 응용하는 부분에 있어서는 아직 어려움을 느끼고 있다는 것을 알 수 있었다.

이번 연구에서는 연구 대상이 중학생으로 한정되었고 8주간의 프로그램 과정이 끝난 이후 프로그램에 대한 자신감이 향상되었는지 다른 교과목에 컴퓨팅 사고력을 적용하여 문제를 해결할 수 있는지에 대한 사후 만족도 조사를 통해 프로그래밍에 대한 자신감은 높아졌지만 다른 교과목에 컴퓨팅 사고력을 적용하는 부분에서는 다소 어려움이 있다는 것을 알 수 있었다. 그러나 교육 프로그램 적용 전후에 대한 효과 비교나 실험집단과 비교집단 간의 비교 등이 이루어지 않아 향후 연구에서는 보다 객관적인 검증이 필요하다고 판단된다.

따라서 후속 연구에서는 다양한 집단을 연구 대상으로 선정하여 수업 전후 비교, 실험집단과 비교집단 간의 비교 연구를 실시하고 신뢰성 있는 측정 도구를 통해 객관적인 효과 검증을 할 것이다.

프로그래밍 교육을 통한 소프트웨어 교육이 중요하게

강조되고 있지만 여전히 현장에서 학습자들은 프로그래밍 교육에 어려움을 느끼고 있다. 따라서 본 논문에서 제안하는 컴퓨팅 사고력을 적용한 파이썬 교육 프로그램이 비전 공자나 프로그래밍 초보 학습자들이 프로그래밍 교육에 대한 거부감 없이 쉽고 재미있게 프로그래밍 교육을 하고 자신감을 가질 수 있는 기회를 제공할 것으로 기대한다.

REFERENCES

- [1] Youn-Gjik Km, Seong-Sik Kim, "A Study on Computational Thinking based Test-Driven Problem Solving Learning Model," The Journal of Korean Association of Computer Education, Vol. 22, No.6, pp. 43-55, Nov 2016. DOI : 10.32431/kace.2019.22.6.005
- [2] Chul-Hyun Lee, "Development of Computational Thinking based Problem Solving Model(CT-PS Model) for Software Education," The Journal of Korean Practical Arts Education, Vol. 22, No.3, pp. 97-117, August 2016,
- [3] Hyeon-Mi Park, Jong-Wan Kim, "Educational measures for nurturing SW convergence talent in the era of the 4th industrial revolution," Korea Information Processing Society Review, Vol. 26, No. 1, pp. 64-75, January 2019.
- [4] Jae-Kyung Kim, "The Effect of S/W Experience in Elementary and Middle School Curriculums on Computational Thinking Class in University," The Korean Society for Creative Information Culture, Vol. 5, No. 1, pp. 35-43, April 2019. DOI : 10.32823/jcic.5.1.201904.35
- [5] Jae-Kyung Kim, "Trend Analysis of Software Education Research: Focusing on Programming Language," Journal of Creative Information Culture, Vol. 4, No. 1, pp. 13-23, April 2018. DOI : 10.32823/jcic.4.1.201804.13
- [6] Sook-Young Choi, "Design and Application of an Instructional Model for Flipped learning of Programming Class," Journal of The Korean Association of Computer Education, Vol. 20, No. 4, pp. 27-36, July 2017. DOI : 10.32431/kace.2017.20.4.003
- [7] Soo-Jin Jun, "Design and Effect of Development-Oriented Model for Developing Computing Thinking in SW Education," Journal of The Korean Association of Computer Education, Vol. 20, No. 4, pp. 47-57, July 2017. DOI : 10.14352/jkaie.2017.21.6.619
- [8] Youn-Ju Jeon, Tea-Young Kim, "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 Information Education, Vol. 21, No. 6, pp. 619-627, December 2017. DOI : 10.32431/kace.2017.20.4.005
- [9] Hwan-Soo Kang, Jong-Man Lee, Hee-Chern Kim, "A Study on Computer Programming Education Model based on Python," Journal of Digital Contents Society, Vol.21, No.4, pp. 693-700 Mar, 2020. DOI : 10.9728/dcs.2020.21.4.693
- [10] Eui-Sun Kang, "Structural Software Education Model for Non-majors - Focused on Python," Journal of Digital Contents Society, Vol. 20, No. 12, pp. 2423-2432 December, 2019. DOI : 10.9728/dcs.2019.20.12.2423
- [11] Young-Seok Lee, "Analyzing the effect of software education applying problem-solving learning," Journal of Digital Convergence, Vol. 16, No. 3, pp. 95-100 March, 2018. DOI : 10.14400/JDC.2018.16.3.095
- [12] Ae-Hwa Lee, "Domestic Research Trend Analysis of Computing Thinking," The Korea Contents Association, Vol. 19, No. 8, pp. 214-223, August 2019. DOI : 10.5392/JKCA.2019.19.08.214
- [13] J. M. Wing, "Computational thinking," Communications of the ACM, Vol. 49, No. 3, pp. 33-35, March, 2006. DOI : 10.1145/1118178.1118215
- [14] Ja-Mi Kim. Wing, "Problem-solving ability based on computational thinking," Korea Information Processing Society Review, Vol. 24, No. 2, pp. 13-21, March, 2017.
- [15] Kyung-Mi Lee, "Computational Thinking Education Teaching Method Research for Non-Major Subjects," Korean Journal of General Education, Vol. 13, No. 1, pp. 321-343, February, 2019.
- [16] Eui-Sun Kang. "Structural Software Education Model for Non-majors - Focused on Python," Journal of Digital Contents Society, Vol. 20, No. 12, pp. 2423-2432, December, 2019. DOI : 10.9728/dcs.2019.20.12.2423
- [17] Jung-Sook Sung, Soo-Hwan Kim, Hyeon-Cheol Kim. "Analysis of Art and Humanity Major Learners' Features in Programming Class," The Korean Association of Computer Education, Vol. 18, No. 3, pp. 25-35, May, 2015. DOI : 10.32431/kace.2015.18.3.003
- [18] KERIS, Training Textbooks for Software Leading Teachers, Elementary Schools KERIS, <http://lib.keris.or.kr/search/detail/CAT000000013040>.
- [19] CSTA & ISTE, Computational thinking in K-12 education teacher resource second edition, CSTA & ISTE, https://www.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2.
- [20] Jeong-Min Lee. Eun-Ji Ko, "The Effect of Software Education on Middle School Students Computational Thinking," The Korea Contents Society, Vol. 18, No. 12, pp. 238-250, December, 2018. DOI : 10.5392/JKCA.2018.18.12.238
- [21] M. Berland, U. Wilensky, "Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking," Journal of Science Education and Technology, Vol. 24, No. 5, pp. 628-647, October, 2015. DOI : 10.1007/s10956-015-9552-x
- [22] Young-Ho Seo, Mi-Ryeong Yeom, Jon-Hoon Kim, "Analysis of Effect that Pair Programming Develops of Computational Thinking and Creativity in Elementary Software Education," Journal of The Korean Association of Information Education, Vol. 20, No. 3, pp. 219-234, October, 2016.

- [23] Chul-Hyun Lee, "Effects of Computational Thinking based Real Life Problem Solving Learning on Elementary School Student's Computational Thinking," The Society of Korean Practical Arts Education Research, Vol. 23, No. 4, pp. 91-107 November, 2017. DOI : 10.29113/skpaer.2017.23.4.091
- [24] Hyung-Shin Choi, "Domestic Literature Review on Computational Thinking Development through Software Programming Education," Journal of Educational Technology, Vol. 34, No. 3, pp. 743-774 October, 2018. DOI : 10.17232/KSET.34.3.743
- [25] Eun-Ji Ko, Jeong-Min Lee, "Assessment Process Design for Python Programming Learning," Journal of The Korean Association of Information Education, Vol. 24, No. 1, pp. 117-129, February 2020. DOI : /10.14352/jkaie.2020.24.1.117
- [26] <https://python.org>
- [27] Kang-soo You, Ki-Cheon Hong, Se-min Kim, Soo-Kyoung Choi, "The Analysis of Learning Demotivation according to Gender and Programming Subjects in Programming Class' Students of Liberal Arts," Journal of the Korea Institute of Information and Communication Engineering, Vol. 23, No. 6, pp. 704-710, Jun 2019.

Authors



Min-Kyung Lee received the B.S., M.A. degrees in Computer Science and Computer Education major from Silla University, Korea, in 2000, 2003, respectively and completed the courses for a doctoral program Education &

media technology major from Pusan National University, Korea, in 2010. she is currently a Assistant Professor in the Department of College of Liberal Arts, Silla University. She is interested in computer education, programming education, AI education and big data.