



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

박사학위논문

유아 소프트웨어 교육을 위한  
교육과정 개발 연구

제주대학교 대학원

과학교육학부 컴퓨터교육전공

이경희

2020년 2월

# 유아 소프트웨어 교육을 위한 교육과정 개발 연구

指導教授 趙正元

李景姬

이 論文을 教育學 博士學位 論文으로 提出함

2019年 12月

李景姬의 教育學 博士學位 論文을 認准함

審査委員長	신정우	
委員	김철민	
委員	백남제	
委員	근재동	
委員	조정원	

齊州大學校 大學院

2019年 12月

# A Study on Curriculum Development for Early Childhood Software Education

KyungHee Lee  
(Supervised by professor Jungwon Cho)

A thesis submitted in partial fulfillment of the requirement for the  
degree of Doctor of Philosophy in Education

2019. 12.

This thesis has been examined and approved.

.....  
Thesis director, Jungwon Cho, Prof. Department of Computer Education  
.....

.....  
(Name and signature)

.....  
Date

Major in Computer Education  
Faculty of Science Education  
GRADUATE SCHOOL  
JEJU NATIONAL UNIVERSITY

# 목 차

표 목 차 .....	iii
그림목차 .....	v
국문초록 .....	vi
I. 서론 .....	1
1. 연구의 필요성 및 목적 .....	1
2. 연구의 내용 및 방법 .....	6
II. 소프트웨어 교육 현황 및 동향 .....	9
1. 국내외 소프트웨어 교육현황 .....	9
1) 영국 .....	9
2) 미국 .....	15
3) 그 외 국가 .....	19
4) 해외 소프트웨어 교육의 시사점 .....	24
5) 국내 .....	26
6) 국내 소프트웨어 교육의 시사점 .....	33
III. 유아 소프트웨어 교육과정 개발 .....	39
1. 유아 소프트웨어 교육의 방향 .....	39
1) 유아의 특성 및 교육 .....	40
2) 유아 소프트웨어 교육의 개념 .....	41
3) 유아 소프트웨어 교육의 필요성 .....	42
4) 유아 소프트웨어 교육 교수자 .....	43
5) 유아 소프트웨어 교육 교구 .....	46

2. 교육과정 개발 절차 .....	49
3. 교육과정 구성 .....	50
1) 교육 목적 .....	52
2) 내용체계 .....	57
3. 교육과정 검증 및 결과 .....	64
IV. 교육 프로그램 적용 및 효과 분석 .....	71
1. 연구 방법 .....	71
1) 연구 대상 .....	71
2) 연구 도구 .....	72
3) 검사 도구 .....	80
4) 연구 절차 .....	86
2. 연구 결과와 분석 .....	90
1) 기술 통계 설계 .....	90
2) 유아 소프트웨어 교육 효과성 .....	91
3) 유아 소프트웨어 교육이 컴퓨팅 사고력에 미치는 효과 .....	95
4) 그 외 교육적 효과 .....	100
V. 결론 및 제언 .....	103
참 고 문 헌 .....	106
ABSTRACT .....	116
부    록 .....	120

## 표 목 차

<표 I-1> 연구 방법 및 산출물 .....	7
<표 II-1> 영국 컴퓨팅 교육 성격과 교육 목표 및 내용 .....	11
<표 II-2> 컴퓨팅 교과 Key stage-1 주요 학습 내용 .....	12
<표 II-3> 컴퓨팅 교재 Key stage-1 구성 .....	13
<표 II-4> 미국 K-12 1 단계 교육과정 세부내용 .....	16
<표 II-5> 미국 K-12 1 단계 영역별 학습 내용 .....	17
<표 II-6> 미국 K-12 1 단계 내용체계 .....	18
<표 II-7> 에스토니아 ProgeTiger 교육과정 .....	20
<표 II-8> 핀란드 KOODI2016 교육과정 .....	21
<표 II-9> 인도 1-2 학년 CMC 교육과정 .....	22
<표 II-10> 싱가포르 코딩 교육과정 .....	24
<표 II-11> 해외(유럽) 초등학교 코딩교육 현황 .....	25
<표 II-12> 해외(유럽) 코딩교육과 교육과정 현황 .....	26
<표 II-13> 누리과정 생활도구로서의 미디어 주요내용 .....	27
<표 II-14> 누리과정 미래의 생활도구 주요내용 .....	29
<표 II-15> 실과 소프트웨어 영역 세부 내용 .....	31
<표 II-16> 초등 소프트웨어 교육과정 표준 모델 내용체계 .....	32
<표 II-17> 유아 소프트웨어 교육 선행 연구 .....	34
<표 II-18> 유아 소프트웨어 교육 선행 연구 분류 .....	37
<표 III-1> 교수자 형태 분류 .....	44
<표 III-2> 유아 소프트웨어 교육 교구 분석 .....	47
<표 III-3> 교육 목적 분류표 .....	53
<표 III-4> 교육 목적 분류 정리표 .....	55
<표 III-5> 내용체계 분류표(미국, 영국, 한국 초등) .....	58
<표 III-6> 내용체계 분류표(한국 중학교, ICILS) .....	60

<표 III-7> 내용체계 분류 정리표 .....	61
<표 III-8> 내용체계 및 교육 프로그램 예시 .....	62
<표 III-9> 교육과정 3차 최종 델파이 내용 타당도 .....	66
<표 III-10> 분석결과와 주요 수정 사항 .....	67
<표 III-11> 유아 소프트웨어 교육 목적 .....	68
<표 III-12> 유아 소프트웨어 교육 내용체계 .....	69
<표 IV-1> 소프트웨어 교육 프로그램 .....	72
<표 IV-2> 소프트웨어 교육 프로그램 10차시 .....	74
<표 IV-3> 컴퓨팅 사고력 검사도구 구성 .....	85
<표 IV-4> 사전 교사교육 .....	87
<표 IV-5> 실험 설계 .....	90
<표 IV-6> 소프트웨어 교육 효과성 검사 정규성 검정 .....	91
<표 IV-7> 소프트웨어 교육 효과성 검사 결과(Wilcoxon 부호 순위 검정) .....	92
<표 IV-8> 소프트웨어 교육 효과성 검사 영역별 결과(Wilcoxon 부호 순위 검정) .....	94
<표 IV-9> 소프트웨어 교육 효과성 검사 총점 .....	94
<표 IV-10> 컴퓨팅 사고력 검사 정규성 검정 .....	96
<표 IV-11> 컴퓨팅 사고력 결과 (Wilcoxon 부호 순위 검정) .....	97
<표 IV-12> 컴퓨팅 사고력 컴퓨터과학 영역별 결과(Wilcoxon 부호 순위 검정) 99	
<표 IV-13> 컴퓨팅 사고력 총점 .....	99

## 그림 목 차

[그림 I -1] 연구 절차 .....	6
[그림 II-1] 2015 개정 교육과정(초.중고등학교 정보교과) .....	30
[그림 II-2] 소프트웨어 교육을 통한 인재상 .....	33
[그림 III-1] 교육과정 개발 절차 .....	49
[그림 III-2] 교육 목적 개발 절차 모형도 .....	53
[그림 III-3] 내용체계 개발 절차 모형도 .....	57
[그림 III-4] 교육과정 검증 절차 .....	65
[그림 IV-1] 유아 소프트웨어 교육 프로그램 예시-1 .....	77
[그림 IV-2] 유아 소프트웨어 교육 프로그램 예시-2 .....	78
[그림 IV-3] 질문지법+면접법 .....	83
[그림 IV-4] 연구 진행 및 절차 .....	86
[그림 IV-5] 소프트웨어 교육 효과성 검사 결과 비교 .....	93
[그림 IV-6] 소프트웨어 교육 효과성 검사 총점 비교 .....	95
[그림 IV-7] 컴퓨팅 사고력 검사 결과 비교 .....	98
[그림 IV-8] 컴퓨팅 사고력 검사 총점 비교 .....	100

<국문초록>

# 유아 소프트웨어 교육을 위한 교육과정 개발 연구

이 경 희

제주대학교 대학원 과학교육학부 컴퓨터교육전공

지도교수 조 정 원

본 연구는 만 5세 대상 유아 소프트웨어 교육과정을 개발하고 적용하여 이를 통해 소프트웨어 교육 효과성 검증과 컴퓨팅 사고력 향상에 미치는 효과에 대해 알아보는데 그 목적이 있다.

인공지능 시대는 더 이상 먼 미래가 아닌 현실이며 소프트웨어가 생활의 기반이고 중심이 되는 세상이다. 사회는 급변하고 있고 변화와 함께 삶의 구조도 다양하고 복잡해지고 있다. 미래 사회를 살아갈 유아들은 더욱 다양한 해결해야 할 문제들과 직면할 것이다. 세계 각국에서는 컴퓨팅 사고력과 문제해결력을 길러주기 위해 보다 어린 나이부터 소프트웨어 교육을 시작하기 위한 국가 차원의 연구와 노력을 기울이고 있다. 소프트웨어 교육은 미래 삶을 위해 누구나 배워야 하는 보편교육이 되어야 한다. 만 5세는 발달과 관련된 모든 영역에서 급속한 성장과 변화가 이루어지는 시기이다. 프로그래머나 관련 직업을 가지지 않더라도 소프트웨어 교육은 이 시기에 필수적이다. 가급적 어린 나이에 체험과 놀이 활동을 통해 사고하는 방법을 일깨워 주고 문제를 발견하고 스스로 해결책을 찾아내는 경험을 할 수 있는 교육을 시작하는 것이 좋다. 이러한 내용을 담기 위해서는 단순히 코딩 능력을 기르거나 로봇 활용으로 그치는 교육이 아니라 체험과 다양

한 놀이로 구성된 간접적이고 자연스러운 활동으로 접근할 필요성이 있다. 즉, 경험을 통해 소프트웨어 교육의 궁극적인 목적인 컴퓨팅 사고력을 함양할 수 있는 체계적인 교육과정이 필요하다. 더불어 사교육을 통한 선택적 교육이 아니라 국가 교육과정 내에서 모든 유아들이 교육 받을 수 있는 시스템이 구축 되어야 한다.

이에 본 연구에서는 국가수준의 유아 교육과정과 조화를 이루면서 초중등과 연계가 가능한 교육과정으로 국가 수준에서 시작할 수 있는 체계를 마련하기 위해 만 5세 대상 유아 소프트웨어 교육과정을 개발하였다.

이를 위해 본 연구는 다음과 같은 절차로 진행되었다.

첫째, 국내외 유아 소프트웨어 교육과정, 관련 연구 동향과 문헌 분석, 선행 연구를 조사하고 비교·분석하였다. 이러한 분석은 교육 목적과 내용체계 개발의 토대가 되었다.

둘째, 유아 소프트웨어 교육 목적 및 내용체계로 구성된 교육과정을 개발하여 전문가 검증을 통한 내용 타당도 검사, 델파이 조사를 통해 완성하였다.

셋째, 개발된 소프트웨어 교육과정을 만 5세 유아들에게 적용한 후 소프트웨어 교육 효과성 검사 도구와 컴퓨팅 사고력 검사 도구를 이용하여 연구 결과 도출 및 분석하고 효과를 검증하였다.

이 과정에서 도출된 결과물은 유아 소프트웨어 교육 목적, 내용체계, 교수·학습 방법, 교육 프로그램, 검사 도구이다.

본 연구에서는 다음과 같은 교육적 효과가 검증되었다.

첫째, 소프트웨어 교육 효과성 검사 결과를 분석해보면 본 연구에서 개발한 교육과정은 만 5세 대상 유아 소프트웨어 교육에 효과적인 것으로 나타났다. 교육을 경험하기 전에 비해 교육을 경험한 후 검사의 하위요인인 소프트웨어 교육의 가치에 대한 이해, 태도, 컴퓨팅 사고력 효능감, 흥미가 높아졌으며, 이는 통계적으로 유의미한 차이를 보였다.

둘째, 컴퓨팅 사고력 검사 결과를 분석해보면 본 연구에서 개발한 교육과정은 만 5세 대상 유아 컴퓨팅 사고력 향상에 효과적인 것으로 나타났다. 교육을 경험하기 전에 비해 교육을 경험한 후 컴퓨팅 사고력이 향상되었으며, 이는 통계적으로 유의미한 차이를 보였다.

본 연구의 시사점은 다음과 같다.

첫째, 소프트웨어 교육의 목적과 필요성에 대해 충분한 내용을 담고 있으면서 국가수준의 유아 교육과정과 조화를 이루고 초중등과 연계가 가능한 교육과정을 제시하였다. 이를 바탕으로 개발된 교육 프로그램은 충분한 교육적 효과가 있음이 검증되었다. 앞으로 관련 분야의 후속 연구에서 유아 소프트웨어 교육과 컴퓨팅 사고력의 본질을 이해하는데 참고가 될 수 있을 것이다.

둘째, 유아 수준에 맞는 소프트웨어 효과성 검사 도구와 컴퓨팅 사고력 검사 도구를 제시하였다. 이를 통해 유아 소프트웨어 교육의 효과성과 컴퓨팅 사고력 향상을 입증하였다. 컴퓨팅 사고력 검사는 논리력 검사, 창의력 검사와 구분해서 사용할 필요성이 있다. 본 연구에서 개발한 검사 도구는 추후 관련 연구의 목적에 맞게 수정·보완하여 사용할 수 있으며 컴퓨팅 사고력 자체 검사의 의미와 필요성을 이해하는 데 많은 도움이 될 것이다.

향후 후속 연구를 통해 본 연구에서 개발한 교육과정을 지속적으로 개선하고 가정과의 연계교육, 부모 참여 교육, 현직 유아교사 교육, 예비 유아교사 양성과정 등으로 발전할 수 있기를 기대한다. 더불어 만 5세를 위한 타당도와 신뢰도를 갖춘 양적 컴퓨팅 사고력 검사 도구가 개발된다면 유아 개인의 컴퓨팅 사고력 향상에 대한 효과를 분석 할 수 있을 것이다.

주요어: 유아 소프트웨어 교육, 소프트웨어 교육, 컴퓨팅 사고력, 교육과정, 소프트웨어 교육 효과

# I. 서론

## 1. 연구의 필요성 및 목적

인공지능 시대는 더 이상 먼 미래가 아닌 현실로 다가왔으며 미래 사회를 준비하는 사람들에게 소프트웨어 역량은 그 중요성이 더욱 강조되고 있다. 세계 각국에서는 미래 인재 양성을 통한 국가 경쟁력 확보를 위해 다양한 연구와 노력을 기울이고 있다(고병오, 2016). 소프트웨어가 기반이 되며 일상생활에서 차지하는 비중이 점점 커지는 소프트웨어 중심 사회로의 변화는 소프트웨어 교육에 대한 요구로 이어지고 있다. 이러한 요구에 발맞추어 실생활의 다양한 문제를 발견하고 해결 가능한 형태로 표현하는 사고과정을 통해 해결책을 제시할 수 있는 컴퓨팅 사고 증진 위주로 국가별 소프트웨어 교육과정의 변화 및 교과 편성이 활발히 이루어지고 있다.

소프트웨어 교육 시작 시기는 국가마다 조금씩 다르지만 미국, 영국의 경우 만 5세부터 컴퓨팅 사고력 기반의 소프트웨어 교육을 시작하고 있다. 핀란드, 포르투갈, 그리스, 에스토니아, 싱가포르 등 많은 나라가 소프트웨어 교육을 정규 수업 과정으로 편성해 실시하고 있으며 교육을 시작하는 연령층 또한 점점 낮아지는 추세이다(안상진 & 이영준, 2014). 주요 국가에서 만 5세부터 유아 소프트웨어 교육을 시작하는 이유는 이 시기 발달의 정도가 개인의 전 생애 학습 태도 뿐 아니라 학습능력과 직업역량에까지 큰 영향을 미칠 수 있기 때문이다. 즉, 만 5세 유아는 새로운 학습과 경험을 통한 다양하고 복잡한 사고가 가능하며, 이를 통해 학업성취까지 향상되어지고 성인으로 성장하기 위한 사회적 정서적 기술을 습득할 수 있는 준비가 되어 있는 상태를 갖추고 있다(Torrance, 1965). 프로그래머나 관련 직업을 가지지 않더라도 이 시기의 소프트웨어 교육은 필수적이다.

최근 유아 교육은 미래 인재 역량개발 중심으로 교육과정이 변하고 있다(유아정책연구소, 2012). 이와 더불어 세계 각국에서는 소프트웨어 교육을 정규과정으

로 도입하여 통합적 사고력, 창의적 문제해결 능력을 갖춘 인재양성을 시도하고 있다. 이러한 변화에 발맞추어 한국은 2015 교육과정 개정안이 고시되면서 소프트웨어 교육을 강조하였고 이에 정보 교과를 강화하였다. 2018년 중학교를 시작으로 2019년에는 초등학교까지 확대 실시하고 있다(교육부, 2018a). 그러나 중학교 3년 중 34시간, 초등학교는 5~6학년 실과 과목에 17시간 소프트웨어 교육 내용을 포함시켜 운영하고 있어 교육과정 편제상 충분한 시간을 확보하고 있다고 보기 어렵다. 또한, 교육과정상 초등과 중등의 연계성이 부족하고 소프트웨어 교육을 제대로 할 수 있는 전문교사마저 부족한 실정이다(김갑수, 2016).

유치원의 소프트웨어 교육을 살펴보면 교육부에서 선행학습을 우려해 2016년 9월 전국 유치원 ‘만 3~5세 대상 코딩교육을 금지’한다는 내용의 공문을 각 시도 교육청에 발송하였으며, 2017년 4월 유치원 내 코딩교육 금지를 재차 강조했다(제주특별자치도교육청, 2017). 그러나 보건복지부 산하 어린이집에서는 코딩교육 금지나 권고 등은 검토하지 않고 있고 실제로 방과후 과정을 통해 교육을 진행하고 있는 상황이다. 동일 대상이어도 관계부처에 따라 소프트웨어 교육에 대한 입장을 달리하고 있어 정책적 혼선이 빚어지는 실정이다. 또한, 학부모들의 소프트웨어 교육에 대한 관심은 점점 증가하고 있어 일부 유치원에서는 교육부 권고에도 불구하고 코딩교육을 실시하고 있고 검증되지 않은 교구를 통한 무분별한 교육이 우후죽순 생겨나 사교육시장을 중심으로 점점 입지를 넓혀가고 있다.

일반적인 코딩 교육은 원래의 의미와는 다르게 단순히 코딩 능력, 기능 활용교육으로 블록 조립이나 로봇 활용에 초점을 두고 있기 때문에 컴퓨팅 사고력 증진보다 흥미 위주의 교육이 될 수밖에 없다(이경희 외, 2018a). 활용 교육을 통해 컴퓨팅 사고력이 향상되고 더 좋은 방법으로서의 연결과 새로운 것으로 연결로 이어지지 못하는 문제점이 있다. 소프트웨어 교육의 목적은 단순히 코딩 능력, 로봇 활용능력을 기르는 것이 아니라 주어진 문제를 보다 창의적으로 해결하는 핵심 능력인 컴퓨팅 사고력을 향상시키는 것이다. 즉, 단순히 활용하는 것으로 그치는 지식을 가르치는 것이 아니라 전체적으로 사고하는 능력을 길러주는 것이다.

육아정책연구소(2014) 보고에 따르면 스마트폰을 최초로 이용하는 시기는 만

2.27세로, 만 3세가 되기 전으로 나타났다. 뿐만 아니라 주중 이용 시간은 31.65 분이고, 주말 이용시간은 39.05분으로 주중보다 주말에 더 많이 이용하는 것으로 나타났다(양진희, 2019). 과학기술정보통신부와 한국정보화진흥원(2016)의 스마트폰 과의존 조사 결과를 살펴보면 전체 과의존 위험군 5,810명 중 유아동은 12.4%, 과학기술정보통신부와 한국정보화진흥원(2018)의 조사에서는 유아동의 과의존 위험군이 20.7%로 나타나 유아의 스마트폰 중독이 해마다 가파르게 증가하고 있는 것을 확인할 수 있다(조준오 & 홍광표, 2019). 이러한 상황을 선제적으로 대비하기 위해서라도 더욱 어린 나이부터 제대로 된 소프트웨어 교육이 필요하다. 안전하고 제대로 스마트 기기를 사용할 수 있는 정보윤리 교육, 정보보안 교육, 스마트 기기 과의존 교육이 포함된 유아 소프트웨어 교육을 더욱이 미룰 수가 없는 상황이다.

더불어 유아 소프트웨어 교육도 컴퓨팅 사고력이라는 소프트웨어 교육의 궁극적인 목적 달성을 위한 교육이 되어야 한다. 컴퓨팅 사고력이란 실생활의 다양한 문제를 발견하고 절차적 사고를 통해 분석하고 추상화하여 컴퓨팅 시스템을 이용하여 문제를 해결하는 종합적인 사고 능력이다. 유아 단계에서 이러한 능력을 길러주기 위해 놀이와 체험 속에 자연스럽게 녹아드는 소프트웨어 교육이 필요하다. 유아들은 놀이를 통해 문제를 탐색하고 선택하며 결정해 나가는 과정에서 주변의 환경과 다양한 상호작용을 한다. 이를 통해 폭넓은 학습 기회를 가질 수 있고 학습의 효과도 높일 수 있다. 따라서 유아기 때부터 놀이를 통하여 계열성과 연계성으로 체계적인 경험을 제공해주면 컴퓨팅 사고력의 조직화가 이루어질 수 있으며 문제해결 과정에서 발상하는 창의성과 논리적 사고가 향상 될 수 있다.(김정민 외, 2016) 창의성 발달이라는 관점에서 살펴볼 때, 유아기는 창의성 계발에 있어 무한한 발전 가능성을 지니고 있는 시기이므로 창의성 계발의 최적기이다(이경화, 2006). 기초 단계의 소프트웨어부터 쉽고, 즐겁게 배울 수 있도록 교육하여 소프트웨어 교육을 어려워하지 않고 친숙하게 받아들일 수 있는 교육 환경을 조성해 주는 것이 중요하다(김형숙, 2015).

유아교육에서는 컴퓨터 관련 교육이 유아의 발달적 특성에 적합하지 않다는 입장과 교육적 효과에 대한 검증이 부족하다는 관점, 유아들에게 놀이와 구체적 경험, 자유로운 탐색을 보장해 주어야 한다는 지적 등에 부딪혀 컴퓨터 관련 교

육 도입에 대해 다소 소극적이고 비수용적 자세를 취하였다. 이로 인해 유아교육에 소프트웨어 교육의 시작은 초·중등학교에 비해 다소 늦어지게 되었다(남궁선혜, 2002; 홍혜경, 2002).

최근 들어 유아교육 전문가들이 소프트웨어 교육에 관심을 가지고 활발하게 연구를 진행하고 있으나 현재까지 이루어진 유아 소프트웨어 교육은 컴퓨터 사용의 기술적인 측면에만 치우쳐 관련 지식을 형성하는 내용이 비중 있게 다루어지지 못하였다. 선행 연구들을 살펴보면 단순히 코딩 프로그램 교육, 로봇 활용 교육에 초점을 맞추어 흥미위주의 제한적인 교육에 집중되어 있다. 이러한 이유는 소프트웨어 전문 지식의 한계로 교육 속에 담아낼 수 있는 내용적 한계가 있기 때문이다. 유아교육 관점에서 소프트웨어 교육 연구도 중요하지만, 소프트웨어 교육 관점에서 유아 소프트웨어 교육에 관한 연구는 매우 의미 있는 일이라고 할 수 있다.

유아 소프트웨어 교육은 단순히 흥미 위주의 코딩, 로봇 활용교육으로 접근하고 시작해서는 안 된다. 활용하는 법을 배우는 것을 뛰어넘어 컴퓨팅 사고력을 함양하고 문제해결력을 길러주는 데 바탕이 되어야 한다. 이러한 사고력은 다양한 분야의 경쟁력을 기르는데 도움이 될 것이다. 이러한 이유로 컴퓨팅 사고력이라는 소프트웨어 교육의 궁극적인 목적 달성을 위한 제대로 된 교육과정과 더불어 초·중등과 연계가 가능한 교육이 필요하다. 윤리적 행동을 이해하고 알맞은 소프트웨어를 활용하는 정보윤리 교육을 반드시 포함하여야 한다. 친구들과 함께 정보를 수집하고 공유하는 과정을 통해 문제를 해결해보는 경험이 동반되어야 한다. 일상생활의 사례를 중심으로 구성된 활동을 통해 내용을 경험하고 자연스러운 학습 환경을 조성하여 체험과 놀이 속에 녹아드는 풍부한 교육적 경험이 바탕이 되어야 한다.

본 연구에서는 이러한 내용을 바탕으로 실생활의 다양한 문제를 해결해 보는 경험을 통해 문제해결력과 컴퓨팅 사고력을 향상시키는데 목적을 두고 있다. 적기 교육이 어린 아이들의 학습과 발달에 가져올 수 있는 큰 영향을 감안할 때, 만 5세부터 소프트웨어 교육이 중요하며 필요하다고 제안한다. 소프트웨어 교육은 유아들이 일반적으로 갖추어야 할 기초지식과 보편적으로 길러줘야 하는 능력을 위해 필요한 교육이다. 이를 위해서는 보편교육이 되어야 하며 모든 유아가

경험할 수 있도록 사교육이 아니라 국가 교육과정인 누리과정 내에서 자연스럽게 교육을 받을 수 있는 시스템이 구축되어야 한다. 이에 따라 ‘만 5세 대상 유아 소프트웨어 교육’의 방향을 정립하기 위해 주요국가 동향 분석, 선행 연구 분석, 사례 연구 등을 시행하여 교육목적과 내용체계를 구성하였다 이를 검증하기 위해 전문가 심층 인터뷰를 통한 요구분석, 내용 타당도 검증, 전문가 델파이를 통한 타당성 검증을 거친 교육 목적 및 내용체계로 구성된 교육과정을 개발하였다. 이를 통해 소프트웨어 교육 효과성과 컴퓨팅 사고력 향상 검증으로 만 5세부터 유아 소프트웨어 교육 시작의 가능성과 당위성을 제안하고자 한다.

## 2. 연구의 내용 및 방법

본 연구에서의 연구 내용은 다음과 같다.

첫째, 만 5세 유아 소프트웨어 교육을 위한 교육과정을 개발하기 위해 국내외 소프트웨어 교육과정과 사례 분석, 선행 연구들을 조사하고 그 특징 및 주요 내용들을 비교·분석하여 제시한다.

둘째, 유아 소프트웨어 교육의 목적을 제시하고 이를 달성하기 위한 내용체계와 교육 프로그램 예시를 구성한다.

셋째, 교육 목적, 내용체계, 교육 프로그램 예시로 구성된 교육과정을 개발하여 전문가 검증을 통한 타당도 검사를 한다.

넷째, 개발된 소프트웨어 교육과정 및 교육 프로그램을 만 5세 유아들에게 적용한다.

다섯째, 소프트웨어 교육 효과성 검사 도구와 컴퓨팅 사고력 검사 도구를 개발하고 이를 사용하여 본 연구를 통해 개발한 교육과정의 연구 결과를 도출 및 분석하여 효과를 검증한다.

본 연구에서는 체제적 교수 설계의 한 방법인 ISD(Instructional System Development) 모형의 ADDIE 단계에 따라 [그림 I -1] 연구 절차와 같이 연구를 수행한다. 분석 단계에서는 문헌 연구 분석, 사례 연구 분석을 실시하며, 설계 단계에서는 교육 목적, 내용체계, 교수·학습 방법, 평가도구를 선정한다. 개발 단계에서는 교육과정 개발과 평가에 활용될 도구 개발, 평가 설계 및 수행을 한다. 실행 및 평가 단계에서는 유아를 대상으로 교육 프로그램을 적용하고 학습의 효과 및 결과를 분석한다.



[그림 I -1] 연구 절차

ISD 모형에 따른 연구 방법 및 연구 산출물을 좀 더 구체적으로 살펴보면 <표 I-1>과 같다.

<표 I-1> 연구 방법 및 산출물

단계	연구 내용	연구 방법	산출물
분석	국내외 소프트웨어 교육 연구	문헌 연구	소프트웨어 교육 분석
	선행 연구 분석	사례 연구	유아 소프트웨어 교육 현황표
	교육과정 분석	문헌 연구	교육과정 분석표
설계	교육 목적 선정	문헌 연구	유아 소프트웨어 교육 목적 도출
	내용 체계 선정	문헌 연구	교육 목적에 적합한 내용 체계 추출
	평가도구 선정	조사 연구 문헌 연구	교육적 효과를 검증하기 위한 검사 도구 선정
	교수 방법 선정	조사 연구 문헌 연구	교육에 효과적인 교수 방법 선정
개발	교육과정 개발	개발 연구	교육 목적, 내용체계
	평가도구 개발	개발 연구	소프트웨어 교육 효과성 검사 도구 컴퓨팅 사고력 검사 도구
실행 및 평가	프로그램 적용	실험 연구 조사 연구	적용 효과 및 결과 분석표
	교육 프로그램 적용 결과 분석		

본 연구가 갖는 제한점은 다음과 같다.

첫째, 본 연구에서 개발된 유아 소프트웨어 교육과정에 따른 프로그램은 현실적으로 당장 국가 교육과정으로 도입이 어렵다. 따라서 방과후 과정을 통해 수업을 실시하였다. 그러나 교육기관의 특성에 따라 방과후 과정이나 관련 프로그램을 통한 소프트웨어 교육을 시작으로 추후 국가 교육과정에 도입하기 위한 틀을 마련하기 위한 기초가 될 수 있을 것이다.

둘째, 유아에게 편안하고 익숙한 환경에서 교육을 진행하여야 최적의 효과를 기대할 수 있으므로 담임교사가 직접 교육을 하는 것이 좋다. 그러나 이를 위해

서는 소프트웨어 교육 내용에 대한 장기간 체계적인 교사교육과 더불어 관련 연수가 필요하기 때문에 본 연구에서는 소프트웨어 전문 강사가 수업을 진행하였다. 추후 유아교사 관련 연수 및 예비 유아교사 양성과정 등을 통한 소프트웨어 교육이 선행되어야 할 것이다.

## Ⅱ. 소프트웨어 교육 현황 및 동향

### 1. 국내외 소프트웨어 교육현황

인공지능 시대는 소프트웨어가 중심이 되고 기반이 되는 시대이며 생각보다 빠르게 변화하고 있어 이러한 변화에 준비가 필요하다. 소프트웨어는 우리 생활 곳곳에 깊숙이 자리 잡고 있으며 필수적이고 보편적인 교육이 되어가고 있다. 이에 영국, 미국을 비롯하여 에스토니아, 핀란드, 인도, 싱가포르 등 세계 각국에서는 더욱 어린 나이부터 소프트웨어 교육을 실시하기 위해 컴퓨팅 사고력 기반 소프트웨어 교육 활성화에 집중하고 있다. 이러한 변화에 따른 준비를 위해 우리나라도 만 5세부터 소프트웨어 교육을 시작하기 위한 교육과정 개발을 더욱 미룰 수가 없는 상황이다. 유아 소프트웨어 교육과정 개발을 위해 먼저 국내외 소프트웨어 교육 현황과 연구 동향을 살펴보았다. 국내외 소프트웨어 교육 현황에서는 각국의 소프트웨어 교육과정 중 초등학교 1학년에 해당하는 내용들을 위주로 살펴보았으나 실질적으로 우리나라 만 5세에 해당되는 연령이다. 우리나라에서는 소프트웨어 교육이라고 명명하고 있으나 국가별로 컴퓨팅 교육, 코딩 교육, 컴퓨터과학 교육, 코드 교육 등 다양한 명칭을 사용하고 있었다. 국가별 자세한 내용들을 살펴보면 다음과 같다.

#### 1) 영국

영국은 1992년 국가 교육과정에서 ‘정보 기술’을 11개 필수 교과에 포함하여 ICT 활용교육을 시작하였다. 2014년 9월부터는 G20 국가 중 처음으로 ‘컴퓨팅(Computing)’이라는 교과목으로 만 5세부터 16세까지 전 학년 필수 교과로 지정하여 컴퓨터 교육을 실시하고 있다. 이러한 결정은 미국 뿐 아니라 북유럽, 아시아, 우리나라의 교육과정에 직접적인 영향을 주게 되었다(김홍래 & 이승진, 2013). 영국 교육과정에서 초등학교부터 고등학교까지 전체를 아우르는 필수 교

과목은 4과목인데 그 중 컴퓨팅이 포함되어 있는 걸 미루어 볼 때 중요성의 정도를 파악할 수 있다.

컴퓨팅 교과목의 주요 내용을 살펴보면 컴퓨터과학 개념들과 프로그래밍 연습 과정을 통해 컴퓨팅 사고력을 습득할 수 있도록 하여 실생활의 다양한 문제들을 그 관점으로 보게 하고 새로운 것을 창조(make, create)할 수 있도록 하는 디지털 창조자(maker, creator) 역량에 초점을 두고 있다. 컴퓨팅 교과목의 기본 개념은 학생들이 교육을 통해 컴퓨팅 사고력을 함양할 수 있도록 하는 것이다. 이러한 사고력 함양을 통해 논리적 사고와 정확성을 심어줌으로써 세상을 살아가는 힘, 세상을 긍정적으로 변화시키는 힘을 갖게 한다. 컴퓨팅 사고력은 크게 추상화와 자동화, 두 가지의 핵심요소를 지닌다. 또한 컴퓨팅 사고력은 ‘프로그래밍 또는 코딩’ 연습을 통하여 습득될 수 있다고 하며, 컴퓨터과학의 여러 개념들을 통해 완성시킬 수 있도록 구성되어 있다(김현철, 2015). 컴퓨팅 교과목의 핵심요소는 프로그래밍, 컴퓨팅 사고력, 창의성, 네트워크, 의사소통/협업, 생산성의 총 6가지로 구성되어 있으며 학년이 올라가도 반복되지만 심화되는 특성을 가지고 있다.

컴퓨팅 교육과정은 총 4단계로 이루어져 있으며 주당 1시간씩 수업을 진행하고 있다. 우리나라 만 5세에 해당하는 Key Stage-1(1~2학년, 5~7세)의 교육 성격, 교육 목표와 교육 내용을 자세히 살펴보면 <표 II-1>과 같다(England, 2015). 일반적인 기술 교육과 간단한 프로그램을 작성하는 교육에 그치지 않고 정보 통신 기술에 대한 책임감을 강조하고 있어 어린 나이부터 정보윤리 교육이 중요하다는 것을 시사하고 있다.

또한 영국은 국가 수준의 교육 뿐 아니라 다양한 비영리 교육 지원 단체들의 교육도 체계적으로 잘 이루어지고 있다. 더불어 국가에서는 컴퓨터 교사 뿐 아니라 교육의 활성화를 위해 타교과 교사, 교장, 행정을 담당하는 교직원에게까지 연수 프로그램을 지속적으로 제공하며 컴퓨팅 교육을 위한 지원과 노력을 아끼지 않고 있다.

<표 II-1> 영국 컴퓨팅 교육 성격과 교육 목표 및 내용

구분	내용
교육 성격	<ul style="list-style-type: none"> <li>• 고품질의 컴퓨터 교육은 학생들이 세상을 이해하고 변화시키기 위해 컴퓨팅 사고와 창의력을 사용하도록 한다.</li> <li>• 컴퓨팅은 수학, 과학, 디자인 및 기술과 깊은 관련이 있으며 자연 및 인공 시스템 모두에 대한 통찰력을 제공한다.</li> <li>• 컴퓨팅의 핵심은 학생들에게 정보와 컴퓨터의 원리, 디지털 시스템의 작동 원리, 그리고 프로그래밍을 통해 이 지식을 사용하는 방법을 가르치는 컴퓨터 과학이다.</li> <li>• 이러한 지식과 이해를 바탕으로 학생들은 정보 기술을 사용하여 프로그램, 시스템 및 다양한 콘텐츠를 제작할 수 있다.</li> <li>• 또한 컴퓨팅은 학생들이 미래의 직장과 디지털 세상의 적극적인 참여자로서 적합한 수준에서 디지털로 지식을 습득하고 정보를 활용하고 자신을 표현하고 정보 및 통신 기술을 통해 아이디어를 개발할 수 있도록 한다.</li> </ul>
교육 목표	<ul style="list-style-type: none"> <li>• 추상화, 논리, 알고리즘 및 자료 표현을 포함한 컴퓨터 과학의 기본 원리와 개념을 이해하고 적용할 수 있다.</li> <li>• 컴퓨팅 용어로 문제를 분석할 수 있으며, 그 문제들을 해결하기 위해 컴퓨터 프로그램을 작성하는 실질적인 경험을 충분히 반복할 수 있다.</li> <li>• 새로운 기술 또는 익숙하지 않은 기술을 포함한 정보 기술을 분석적으로 평가하고 적용하여 문제를 해결할 수 있다.</li> <li>• 학생들은 정보 통신 기술에 대해 책임감 있고 유능하며 자신감 있고 창의적인 사용자이다.</li> </ul>
교육 내용	<ul style="list-style-type: none"> <li>• 디지털 장치에서 프로그램으로 구현되어 있는 알고리즘과 정확하고 모호하지 않은 명령어를 통해 실행되는 프로그램에 대해 이해하기</li> <li>• 간단한 프로그램을 작성하고 디버깅하기</li> <li>• 간단한 프로그램이 어떻게 실행될지 예측하고 논리적 추론 사용하기</li> <li>• 디지털 콘텐츠를 생성, 구성, 저장, 조작, 검색할 수 있는 기술을 활용하기</li> <li>• 학교 외에서도 정보기술을 보편적으로 사용하고 있음을 인식하기</li> <li>• 기술을 안전하게 사용할 수 있도록 기술지원을 받을 수 있는 연락처를 확인하기</li> </ul>

영국의 소프트웨어 교육을 위한 컴퓨팅 교과서 Key Stage-1의 주요 학습 내용은 <표 II-2>와 같다(김민자 & 성정숙, 2015). 주요 학습 내용을 자세히 살펴보면 알고리즘, 프로그래밍, 데이터, 컴퓨터 구조로 구분되어 있다. 알고리즘은 일상생활에서 쉽게 접할 수 있는 케이크를 만드는 방법과 연결 지어 학습을 하

고 있고, 프로그래밍은 스토리가 있는 학습 내용을 구성하고 있다. 이처럼 컴퓨터의 기능 뿐만 아니라 알고리즘, 프로그래밍 등 주요 내용은 다양한 경험 위주의 활동으로 다루어져야 할 것이다. 더불어 프로그래밍 활용 교육이 아니라 창의성과 논리적 사고력 신장에 중점을 둔 컴퓨팅 사고력 교육이 이루어져야 한다.

<표 II-2> 컴퓨팅 교과 Key stage-1 주요 학습 내용

구분	주요 학습 내용
알고리즘	<ul style="list-style-type: none"> <li>• 알고리즘은 목표를 달성하기 위한 명령어들의 집합, 정의된 단계로 구성된 지시문들의 집합(케이크를 만들 때 ‘어떻게’에 해당하는 부분)</li> <li>• 알고리즘은 단순한 형식으로 표현(스토리보드와 서술)</li> <li>• 알고리즘으로 모든 활동을 묘사할 수 있음</li> <li>• 컴퓨터에게 설명할 때 어떤 단계들은 작은 단계들로 만들어 질 수 있으므로 단계들의 반복에 대해 설명</li> </ul>
프로그래밍	<ul style="list-style-type: none"> <li>• 컴퓨터는(프로세서에 의해 제어되는 모든 장치를 의미하며 프로그래밍이 가능한 장난감, 휴대폰, 게임 콘솔 및 PC들 포함) 명령어들의 조합으로 제어</li> <li>• 컴퓨터 프로그램은 이야기의 서술부분과 같고 컴퓨터는 해설자의 지시에 의해 작업을 수행함</li> <li>• 특정 작업은 컴퓨터에 의한 프로그램을 생성</li> <li>• 컴퓨터는 일반적으로 입력을 받고, 저장된 명령어의 조합에 따라 결과 생성</li> <li>• 프로그램은 반복된 명령문을 포함</li> </ul>
데이터	<ul style="list-style-type: none"> <li>• 정보는 다양한 형태로 저장되고 전달 (숫자, 텍스트, 소리, 이미지, 비디오 등)</li> <li>• 컴퓨터는 정보를 저장하기 위해 2진 스위치(on/off) 사용</li> <li>• 2개의 응답(yes/no)은 직접적으로 유용한 정보(참석, 결석)를 제공하거나 의사결정을 위해 사용</li> </ul>
컴퓨터 구조	<ul style="list-style-type: none"> <li>• 컴퓨터는 저장된 명령어들의 조합을 사용하는 전자 기기</li> <li>• 컴퓨터는 일반적으로 입력으로 받아들이고, 출력을 생성</li> <li>• 우리 생활 및 주변 많은 장치에 컴퓨터를 포함하고 있음</li> </ul>

<표 II-3>은 영국의 컴퓨팅 교재의 내용이다. 각 학년의 컴퓨팅 교재는 교사용 교재와 단원 포스터, CD-ROM으로 구성되어 있다(Berry, 2013; Berry, 2014). 교사용 교재에는 포괄적인 단계별 지시사항과 평가 지침이 포함되어 있다. 포스터에는 각 단원을 개략적으로 설명하고, 단원별 활동에 참여하도록 유도할 수 있

게 되어있으며 해당 단원에 대한 단계별 지침과 반대쪽에는 그 단원을 보충할 수 있는 추가 정보나 지침이 나와 있어 교실에서 활용이 가능하다. CD-ROM에는 각 단원을 지원하는 형식과 예시, 음향 효과 및 사진, 편집 가능한 경과 보고서, 학생 자기 평가 정보 등의 평가지침이 나와 있다. 이처럼 영국 컴퓨팅 교재에는 수업 내용 뿐 아니라 짝이나 조 편성에 관련된 세심한 내용까지 포함하고 있다. 교육과정 개발과 더불어 교사의 부담을 덜어줄 수 있으면서 교육 프로그램 운영을 도와줄 수 있는 교재 개발의 중요성을 시사하고 있다. 또한, 각 단원은 스토리가 있는 교육 내용으로 구성되어 있고 비봇을 활용하는 단원에서 장난감이라는 단어를 통해 친근하게 놀이 개념으로 교육에 접근할 수 있도록 한 것을 살펴볼 수 있다. 다양한 피지컬 도구를 활용하여 놀이위주의 소프트웨어 교육을 시행하고 있다. 협업을 상당히 강조하고 있으며 교재 앞부분 연계과목을 표시하여 타 교과와의 연계성과 더불어 융합교육을 중요시하고 있는 걸 볼 수 있다(이경희 외, 2018b).

<표 II-3> 컴퓨팅 교재 Key stage-1 구성

단원	학습 목표	컴퓨팅 학습 프로그램
1.1 우리는 보물 사냥꾼  프로그래밍 할 수 있는 장난감 사용하기	<ul style="list-style-type: none"> <li>명령을 순서대로 입력하여 프로그래밍 할 수 있는 장난감을 조종하는 원리를 이해할 수 있다.</li> <li>명령의 순서대로 알고리즘을 생성하고 저장할 수 있다.</li> <li>장난감이 알고리즘을 따르도록 프로그램을 작성할 수 있다.</li> <li>프로그램의 오류를 찾아내서 제거할 수 있다.</li> <li>프로그램이 어떻게 수행될지 예측할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>알고리즘이 무엇인지, 알고리즘이 디지털 기기에 프로그램으로 어떻게 적용되는지 이해한다. 또한 프로그램은 분명하고 명확한 명령에 따라 실행된다는 것을 이해한다.</li> <li>간단한 프로그램을 만들고, 오류를 찾아내서 제거한다.</li> <li>논리적으로 추론하여 간단한 프로그램의 동작을 예측한다.</li> <li>학교 밖에서 일반적으로 정보 통신 기술이 어떻게 활용되고 있는지 인지한다.</li> </ul>

단원	학습 목표	컴퓨팅 학습 프로그램
1.2 우리는 TV 요리사  요리 순서 녹화하기	<ul style="list-style-type: none"> <li>알고리즘처럼 과정을 단순하고 명확한 단계로 나눌 수 있다.</li> <li>비디오 카메라의 다양한 기능을 사용할 수 있다.</li> <li>비디오 카메라를 사용하여 동작을 녹화할 수 있다.</li> <li>협동 능력을 기를 수 있다.</li> <li>과제 수행 결과를 토론하고 개선할 점을 찾을 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>알고리즘이 무엇인지, 알고리즘이 디지털 기기에 프로그램으로 어떻게 적용되는지 이해한다. 또한 프로그램은 분명하고 명확한 명령에 따라 실행된다는 것을 이해한다.</li> <li>목적에 따라 디지털 콘텐츠를 생성하고, 분류하고, 저장하고, 처리하고, 불러오는 기술을 사용한다.</li> <li>학교 밖에서 일반적으로 정보 통신 기술이 어떻게 활용되고 있는지 인지한다.</li> <li>논리적으로 추론하여 간단한 프로그램의 동작을 예측한다.</li> </ul>
1.3 우리는 화가  전자책 삽화 그리기	<ul style="list-style-type: none"> <li>인터넷을 안전하게 사용하여 삽화에 대한 아이디어를 검색할 수 있다.</li> <li>컴퓨터로 이미지를 만들고 수정할 때 적합한 그래픽프로그램을 선택하여 사용할 수 있다.</li> <li>물감과 종이를 사용하여 그림을 그리는 것과 컴퓨터를 사용하여 그림을 그리는 것이 어떻게 다른지 이해할 수 있다.</li> <li>특정 목표를 위한 삽화를 그릴 수 있다.</li> <li>작업 내용을 저장하고, 불러오고, 수정하는 방법을 인지할 수 있다.</li> <li>작업 결과를 검토해 보고 피드백을 반영하여 수정할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>목적에 따라 디지털 콘텐츠를 생성하고, 분류하고, 저장하고, 처리하고, 불러오는 기술을 사용한다.</li> <li>학교 밖에서 일반적으로 정보 통신 기술이 어떻게 활용되고 있는지 인지한다.</li> <li>개인 정보를 보호하면서 안전하고 책임감 있게 기술을 사용한다. 인터넷이나 기타 온라인 플랫폼에 접속이나 콘텐츠에 이상이 있을 때 어디에 도움을 받아야 할지 인지한다.</li> </ul>
1.4 우리는 수집가  웹에서 이미지 찾기	<ul style="list-style-type: none"> <li>웹에서 사진을 검색하여 사용할 수 있다.</li> <li>부적절한 사진을 발견했을 경우의 대처 방법을 인지할 수 있다.</li> <li>“예/아니요” 질문을 기반으로 이미지를 분류할 수 있다.</li> <li>명확한 규칙에 따라 두 가지 이상의 집합으로 이미지를 분류할 수 있다.</li> <li>기준에 따라 이미지를 순서대로 정렬할 수 있다.</li> <li>이미지에 대하여 “예/아니요” 질문으로 묻고 답할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>알고리즘이 무엇인지, 알고리즘이 디지털 기기에 프로그램으로 어떻게 적용되는지 이해한다. 또한 프로그램은 분명하고 명확한 명령에 따라 실행된다는 것을 이해한다.</li> <li>목적에 따라 디지털 콘텐츠를 생성하고, 분류하고, 저장하고, 처리하고, 불러오는 기술을 사용한다.</li> <li>개인 정보를 보호하면서 안전하고 책임감 있게 기술을 사용한다. 인터넷이나 기타 온라인 플랫폼에 접속이나 콘텐츠에 이상이 있을 때 어디에 도움을 받아야 할지 인지한다.</li> <li>학교 밖에서 일반적으로 정보 통신 기술이 어떻게 활용되고 있는지 인지한다.</li> </ul>

단원	학습 목표	컴퓨팅 학습 프로그램
1.5 우리는 이야기꾼  오디오북 제작하기	<ul style="list-style-type: none"> <li>• 녹음기를 사용하여 소리를 녹음할 수 있다.</li> <li>• 컴퓨터에 소리를 저장할 수 있다.</li> <li>• 조별 활동을 통해 협동 능력을 함양할 수 있다.</li> <li>• 종이책과 오디오북의 차이점을 이해할 수 있다.</li> <li>• 정보 통신 기술 활용에 관해 토론할 수 있다.</li> <li>• 청중에게 녹음한 내용을 들려줄 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• 목적에 따라 디지털 콘텐츠를 생성하고, 분류하고, 저장하고, 처리하고, 불러오는 기술을 사용한다.</li> <li>• 학교 밖에서 일반적으로 정보 통신 기술이 어떻게 활용되고 있는지 인지한다.</li> <li>• 안전하고 책임감 있게 기술을 사용한다.</li> </ul>
1.6 함께 축하해요  디지털 카드 만들기	<ul style="list-style-type: none"> <li>• 타이핑하여 텍스트 서식을 만들면서 기본적인 키보드 사용 능력을 익힐 수 있다.</li> <li>• 기본적인 마우스 조작 기술을 익힐 수 있다.</li> <li>• 웹에서 이미지를 검색하고 선택할 수 있다.</li> <li>• 파일을 저장하고 불러올 수 있다.</li> <li>• 텍스트와 이미지를 합칠 수 있다.</li> <li>• 과제 수행 결과를 토론하고 개선할 점을 찾을 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• 목적에 따라 디지털 콘텐츠를 생성하고, 분류하고, 저장하고, 처리하고, 불러오는 기술을 사용한다.</li> <li>• 학교 밖에서 일반적으로 정보 통신 기술이 어떻게 활용되고 있는지 인지한다.</li> <li>• 개인 정보를 보호하면서 안전하고 책임감 있게 기술을 사용한다. 인터넷이나 기타 온라인 플랫폼에 접속이나 콘텐츠에 이상이 있을 때 어디에 도움을 받아야 할지 인지한다.</li> </ul>

## 2) 미국

미국은 국가수준의 교육과정 없이 국가수준의 지침을 제시하고 각 주마다 상이한 교육과정을 운영하고 있다.

소프트웨어 교육의 목적인 Computational Thinking은 Papert(1996)에 의해 개념이 도입 되었으며, Wing(2006)을 통해 알려졌다. 이후 CSTA(Computer Science Teachers Association)와 IEEE(Institute of Electrical and Electronics Engineers) 및 ACM(Association for Computing Machinery) 등은 컴퓨팅 사고력을 향상시킬 수 있는 교육과정의 편성 및 구성을 위한 노력들을 꾸준히 하고 있다(CSTA & ISTE, 2011).

미국은 컴퓨팅 사고력을 모든 교과와 통합하여 지도하는 것이 필요하다고 인식하고 이에 따른 방안을 고려하고 있다. 또한, 컴퓨팅 사고력에서 가장 중요한 개념으로 도입한 것은 문제해결능력이다. 문제해결능력은 학년별로 다른 교육 내

용과 방법을 적용해야하기 때문에 초등학교 저학년부터 체계적으로 지도하는 것이 필요하다(김홍래 & 이승진, 2013). <표 II-4>는 미국의 K-12 컴퓨터과학 교육과정의 단계별 내용 중 우리나라의 만 5세에 해당하는 1단계(K-3) Computer Science and Me의 세부 내용이다.

<표 II-4> 미국 K-12 1단계 교육과정 세부내용

단계	세부 내용
1단계 Computer Science and Me	<ul style="list-style-type: none"> <li>• CT에 관한 간단한 아이디어와 기본적인 정보기술을 통합하여 컴퓨터과학의 기초 개념 소개</li> <li>• 이 교육과정으로부터 도출된 학습 경험은 컴퓨팅이 학생들의 삶의 중요한 부분임을 알 수 있도록 깨우치고 관심을 유도하고 도와주어야 함</li> <li>• 학습 경험은 능동적인 학습, 창작, 탐구 활동에 초점을 맞추어 설계되어야 하고 사회과학, 언어, 수학 및 과학 등의 다른 교과 내에 포함되도록 하여야 함</li> </ul>

미국 정보과학 교육과정의 내용은 컴퓨팅 사고력(Computational Thinking), 협력(Collaboration), 컴퓨팅 실체와 프로그래밍(Computing Practice and Programming), 컴퓨터와 통신장치(Computer and Communications Devices), 커뮤니티, 글로벌 및 윤리적 영향(Community, Global and Ethical Impacts) 5개의 영역을 학년과 수준에 맞춰 구성하였다. 우리나라 만 5세에 해당하는 CSTA K-12 중 1단계 영역별 학습내용은 <표 II-5>와 같다. 이 내용을 자세히 살펴보면 놀이나 교육용 소프트웨어 활용을 통해 컴퓨터에 대한 친밀감을 형성하여 유용하게 사용할 수 있는 도구임을 인식할 수 있도록 하고 있으며 기술 및 소프트웨어를 안전하게 사용할 수 있는 디지털 시민의식을 기르기 위한 내용으로 구성되어 있음을 알 수 있었다. 기능과 활용적 측면이 아니라 컴퓨터에 대한 친밀감 형성을 기본으로 이 시기의 기술에 대한 책임의식을 길러주는 교육과 정보윤리 교육이 의미가 있음을 시사하고 있다.

<표 II-5> 미국 K-12 1단계 영역별 학습 내용

영역	학습 내용
Computational Thinking(CT)	<ul style="list-style-type: none"> <li>• 나이에 적합한 문제 해결을 위하여 기술적 자원(퍼즐, 논리적 사고 프로그램)을 활용</li> <li>• 단계별 방법으로 생각, 아이디어, 이야기 등을 표현하기 위하여 쓰기 도구, 디지털 카메라, 그리기 도구를 활용</li> <li>• 정보를 쓸모 있는 순서로 배열(정렬)하는 방법을 이해(컴퓨터 사용하지 않기, 생일로 학생들을 정렬하기)</li> <li>• 소프트웨어는 컴퓨터의 동작을 제어하기 위하여 만들어진 것임을 이해</li> <li>• 0과 1이 정보를 표현하기 위하여 사용될 수 있음을 실연</li> </ul>
Collaboration (CL)	<ul style="list-style-type: none"> <li>• 교사 및 가족, 동료로부터 도움을 받아 다른 사람과 전자적으로 정보를 수집하고 의사소통하기</li> <li>• 정보 기술(technology)을 사용하여 동료, 선생님, 제3자와 협력적, 협동적으로 공부하기</li> </ul>
Computing Practice and Programming (CPP)	<ul style="list-style-type: none"> <li>• 나이에 적합한 과제를 수행하기 위하여 필요한 기술적 자원을 활용</li> <li>• 멀티미디어를 교과학습을 위하여 사용(교육용 소프트웨어 등)</li> <li>• 선생님, 가족, 동료학습자와 함께 발달 단계에 적절한 멀티미디어 자료를 만들기</li> <li>• 단순한 동작을 수행하는 데 필요한 일련의 명령어를 구성(거북 명령)</li> <li>• 컴퓨팅과 기술을 사용하는 일들을 확인하기</li> <li>• 마인드맵 도구를 이용하여 정보를 수집하고 조직하기</li> </ul>
Computers and Communications Devices(CD)	<ul style="list-style-type: none"> <li>• 표준 입출력 장치를 사용하여 컴퓨터 및 관련 장치를 조작하기</li> </ul>
Community, Global, and Ethical Impacts(CI)	<ul style="list-style-type: none"> <li>• 기술 및 소프트웨어의 사용에 있어 책임 있는 디지털 시민의식을 기르기</li> <li>• 기술을 사용하는 동안 긍정 및 부정적인 사회적, 윤리적 행동을 확인</li> </ul>

<표 II-6>은 K-12의 1단계에 해당하는 내용체계이다. 앞에서 살펴본 수직적 연계성과 수평적 교육 분야를 통합하여 제시해 학습자의 학습 단계에 따라 각 교육 분야에서 익힐 수 있는 기능과 습득할 수 있는 능력에 대하여 정의하고 있다. 우리나라 만 5세에 해당하는 내용이지만 변수나 제어, 모듈성 등의 어려운 영역을 표시 하고 있다. 그러나 의미를 가르치는 내용이 아닌 상당히 구체적이고

체계적인 내용으로 구성되어 있는 것을 볼 수 있어 국내에 적용할 수 있는 만 5세 소프트웨어 교육의 수준과 내용을 파악할 수 있다.

<표 II-6> 미국 K-12 1단계 내용체계

Concept	Level 1A (Ages 5-7, K~G2)			
Practice	P1. 포괄적인 컴퓨팅 문화의 조성 P2. 컴퓨팅을 바탕으로 하는 협업	P3. 계산 문제의 인식과 정의 P4. 추상화의 수행과 활용	P5. 계산을 통한 산출물 만들기 P6. 계산을 통한 산출물 테스트 및 개선	P7. 컴퓨팅을 바탕으로 하는 소통
Computing Systems	Devices	1A-CS-01 적절한 소프트웨어 선택하고 활용하여 다양한 작업을 수행하고, 사용자가 사용하는 기술에 대한 요구 사항과 선호도가 다르다는 점을 인식할 수 있다. (P1.1)		
	Hardware & Software	1A-CS-02 적절한 용어를 사용하여 컴퓨팅 시스템(하드웨어)의 일반적인 물리적 구성 요소의 기능을 구분하고 설명할 수 있다. (P7.2)		
	Troubleshooting	1A-CS-03 정확한 용어를 사용하여 기초적인 하드웨어 및 소프트웨어 관련 문제 상황을 설명할 수 있다. (P6.2, P7.2)		
Computing Systems	Cyber security	1A-NI-04 암호의 개념과 필요성을 설명하고 강력한 암호를 사용하여 장치와 정보를 무단 접근으로부터 보호할 수 있다. (P7.3)		
Data & Analysis	Storage	1A-DA-05 컴퓨팅 장치를 사용하여 정보를 저장, 검색, 인출, 수정 및 삭제할 수 있고, 데이터로서 저장된 정보를 정의할 수 있다. (P4.2)		
	Collection, Visualization & Transformation	1A-DA-06 동일한 데이터를 수집하고 다양한 방법을 이용하여 시각화하여 표현할 수 있다. (P7.1, P4.4)		
	Interface & Models	1A-DA-07 표, 그래프 등을 통해 시각화된 데이터 속에서 패턴을 발견하고 설명할 수 있다. (P4.1)		
Algorithms & Programming	Algorithms	1A-AP-08 작업을 완료하기 위한 알고리즘(단계별 명령 집합)을 작성하고 수행함으로써 일상의 과정을 모델링할 수 있다. (P4.4)		
	Variables	1A-AP-09 정보를 표현하는 숫자나 다른 기호를 이용하여 프로그램이 어떻게 데이터를 저장하고 조작하는지를 모델링할 수 있다. (P4.4)		
	Control	1A-AP-10 아이디어를 표현하거나 문제를 해결하기 위해 순차 구조와 간단한 반복 구조를 활용하여 프로그램을 개발할 수 있다. (P5.2)		
	Modularity	1A-AP-11 문제를 해결하기 위해 필요한 절차를 정교한 명령 순서로 분해할 수 있다. (P3.2)		
	Program Development	1A-AP-12 프로그램의 이벤트, 목표 및 예상 결과의 순서를 설명하는 계획을 작성할 수 있다. (P5.1, P7.2) 1A-AP-13 프로그램 개발 과정에서 다른 사람들의 아이디어와 창작물을 사용할 때 적절하게 표시할 수 있다. (P7.3) 1A-AP-14 순차 구조 및 단순 반복 구조가 포함된 알고리즘 또는 프로그램의 오류를 디버깅 할 수 있다. (P6.2)		
Impact of Computing	Culture	1A-IC-16 새로운 컴퓨팅 기술이 구현되거나 채택되기 전후의 우리 삶을 비교할 수 있다. (P7.0)		
	Social Interactions	1A-IC-17 온라인에서 다른 사람을 존중하고 책임감 있게 작업할 수 있다. (P2.1)		
	Safety, Law & Ethics	1A-IC-18 로그인 정보를 타인에게 알려주지 않고, 적절하게 로그오프할 수 있다. (P7.3)		

### 3) 그 외 국가

영국, 미국 이외 국가들의 소프트웨어 교육에 대해 추가적으로 살펴보기 위해 우리나라 만 5세에 해당하는 초등학교 1학년 교육과정 또는 유치원 과정을 살펴보았다. 소프트웨어 교육을 비교적 체계적으로 잘 시행하고 있고 지속적으로 연구를 하고 있다는 평가를 받고 있는 에스토니아, 핀란드, 인도, 싱가포르 네 개의 국가들을 선정해 살펴보고자 한다. 일반적으로 컴퓨터 활용 교육뿐만 아니라 프로그래밍 교육을 하고 있었고 발달에 따라 교수학습을 위한 세부적인 내용을 국가 수준에서 제시하고 있음을 알 수 있었다. 이처럼 소프트웨어 교육 시작 단계부터 사교육이 아니라 국가 수준의 교육과정이 필요하다고 할 수 있다.

#### (1) 에스토니아

에스토니아는 국가 주도로 2012년 'ProgeTiger'라는 프로젝트를 통해 소프트웨어 교육과정을 수립하였다. ProgeTiger는 Programming과 Tiger가 합쳐진 용어로써 즐거운 프로그래밍이라는 의미가 있다. ProgeTiger에서는 기존의 정보 활용교육 및 기술 교육에서 벗어나 문제해결을 통한 창의성 신장과 융합적 사고력에 초점을 두고 있다(배영권 & 신승기, 2017). ProgeTiger에서 제시하고 있는 교수학습 목표는 크게 4가지로 구성되어 있다. 첫째, 학생들의 논리적 사고, 창의성, 수학적 사고능력을 기른다. 둘째, 프로그래밍은 즐거운 활동이며 누구나 할 수 있는 것이다. 셋째, 일상생활 속의 실제적 문제를 해결하는 활동을 통해 프로그래밍의 기초를 가르친다. 넷째, 학생들의 발달에 따른 적절한 프로그래밍 언어를 가르친다(HISTA, 2013). 이처럼 교수학습 목표를 통해 문제해결력을 신장시키기 위한 활동을 제시하고 있음을 살펴볼 수 있으며, 나이와 학년을 고려한 계열화된 교육과정을 구성하여 제시할 필요성이 있음을 알 수 있다.

1~4학년에 해당하는 소프트웨어 교육과정을 살펴보면 <표 II-7>과 같다. 초등학교 저학년에 해당하는 단계로서 프로그래밍에 친숙해지는 내용을 우선하고 있으며, 이를 위해 키보드와 마우스를 사용하는 방법을 배우고 문제해결 과정을 게임을 통해 친숙해질 수 있도록 하였다. 또한, 그래픽 기반의 프로그래밍 언어를 사용하여 문제해결 과정을 프로그래밍으로 간단히 구현해보도록 하였다. 에스토니아의 소프트웨어 교육 내용에는 그래픽 기반 프로그래밍 언어인 Kodu, Logo, Scratch를 포함하고 있었다. 프로그래밍 방법에 대한 교육이 아니라 프로

그래밍을 통한 컴퓨팅 사고력 향상과 연계 가능한 국가 수준의 교육과정 개발이 중요하다는 것을 시사하고 있다.

<표 II-7> 에스토니아 ProgeTiger 교육과정

구분	교육 내용
교육 목표	<ul style="list-style-type: none"> <li>• 학생들의 논리적 사고, 창의성, 수학적 사고능력을 기른다.</li> <li>• 프로그래밍이 즐거운 활동이며 누구나 할 수 있음을 제시한다.</li> <li>• 일상생활 속의 실제적 문제를 해결하는 활동을 통해 프로그래밍의 기초를 가르친다.</li> <li>• 학생들의 발달에 따른 적절한 프로그래밍 언어를 가르친다.</li> </ul>
1~4학년	<ul style="list-style-type: none"> <li>• 프로그래밍에 친숙해지기</li> <li>• 키보드와 마우스를 사용하는 방법을 배우기</li> <li>• 논리적 추론과 사고과정이 포함된 게임을 통해 친숙해지기</li> <li>• 그래픽 기반의 프로그래밍 언어 사용하기</li> <li>• Kodu, Logo, Scratch 등 프로그래밍 언어 사용하기</li> </ul>
프로그래밍 언어	<ul style="list-style-type: none"> <li>• Lego WeDo, Kodu, Logo, Scratch</li> </ul>

(2) 핀란드

핀란드는 선진국으로 나아가기 위한 다양한 노력의 일환으로 코딩을 통한 소프트웨어 교육을 통해 인재 양성을 목표로 하고 있다. 유럽 국가 중에서는 처음으로 알고리즘 사고력 함양을 위한 융합교육을 초등학교 1학년부터 실시하고 있다(허희옥 & 서정희, 2017). 2016년부터 필수 교육과정으로 운영하고 있으며, KOODI2016이라는 명칭의 소프트웨어 교육을 위한 세부 계획이 수립되어 전 학년 전 과목에 접목하여 문제해결력 중심의 융합교육 기반 소프트웨어 교육을 시행하고 있다(신승권 & 배영권, 2015). 소프트웨어 교육 확산을 위해 지속적인 교사 연수와 학생들을 위한 환경을 조성하고 있다(LUMA Center, 2014). 생각하는 방법과 문제해결력을 통한 창의성을 기르는 것과 프로그래밍 방법을 통해 Computational Thinking과 문제해결능력을 기르는 것을 교육의 목적으로 하고 있다.

핀란드 1~2학년에 해당하는 KOODI2016 교육과정을 살펴보면 <표 II-8>과

같다(Koodi2016, 2014). 프로그래밍 자체 교육 보다는 실생활의 많은 부분이 프로그래밍과 관련이 있음을 이해하고 이에 대한 흥미를 지속적으로 갖도록 하는 교육을 진행하고 있다. 핀란드의 소프트웨어 교육은 학년에 따라 활용해야 할 프로그래밍 언어의 범위와 기준을 명확하게 제시하고 있으며, 학습자의 인지적 발달에 따라 교수학습을 위한 세부적인 내용을 국가 수준에서 제시하고 있다. 위계성을 가지고 연결이 가능한 국가 수준의 교육과정 수립이 필요하고 학년별 프로그래밍 언어의 선택 기준과 선정이 필요하다는 것을 시사하고 있다.

<표 II-8> 핀란드 KOODI2016 교육과정

구분	교육 내용
교육 목적	<ul style="list-style-type: none"> <li>• 생각하는 방법과 문제해결력을 통해 창의성을 기르는 것이다</li> <li>• 프로그래밍 방법을 통해 Computational Thinking과 문제해결능력을 기르는 것이다</li> </ul>
1~2학년	<ul style="list-style-type: none"> <li>• 프로그래밍이 명령어를 컴퓨터에게 전해주는 것임을 이해</li> <li>• 놀이를 통하여 자연스럽게 문제해결 방법을 익히기</li> <li>• 프로그래밍 전략과 모델에 긍정적이며 즐겁고 열정적으로 노출</li> </ul>
프로그래밍 언어	<ul style="list-style-type: none"> <li>• CS Unplugged</li> <li>• Robottileikki</li> <li>• Computer Science-a-Box:Unplug Your Curriculum</li> <li>• Computer Science for Fun</li> </ul>

### (3) 인도

인도는 ‘즐거운 컴퓨터 교육과정’이라는 의미를 가진 CMC(Computer Masti Curriculum)라는 명칭의 국가수준의 컴퓨터 교육과정이 있으며 유치원부터 고등학교까지 컴퓨터 과학 교육을 위한 교육과정이 체계적으로 편성되어 있다. 2007년 3월 첫 번째 CMC가 개발되었으며, 이를 기반으로 교과서가 개발되었다(신승기 & 배영권, 2015). 이후 2013년부터 필수교과로 도입해 현재까지 체계적으로 운영하고 있다. 우리나라의 만 5세에 해당하는 초등학교 1학년은 주당 1시간씩 교육을 받고 있다. 컴퓨팅 사고력의 중요성을 강조하며 정부 차원 주도외부 전문가 집단 참여로 교육과정 개발과 컴퓨터과학 교육을 확산시킬 수 있었다. 인도의 컴퓨터과학 교수학습 계획은 상당히 자세하게 설명되어 있고 각 학년에 필

요한 주제와 하위 주제로 구분하여 각 주제에 대한 학습목표와 추천하는 교수학습 방법이 자세하게 정의되어 있다(Sridhar, 2013).

교육의 목적을 살펴보면 첫째, 컴퓨터 활용이 아니라 컴퓨터에 대한 유창성을 길러야 한다(김자미, 2014). 소프트웨어를 활용하는 교육이 아니라 컴퓨터를 기반으로 다양한 컴퓨팅 사고를 할 수 있도록 하는 것을 의미한다. 둘째, 컴퓨터와 관련된 내용을 완벽히 익히는 것이 아니라 이를 통한 생각의 방법과 기술을 개발해야 한다. 생각하는 방법은 문제해결력을 위한 기본 과정이며 이 과정에 기술을 활용하고 소통하는 과정에 의미를 부여한다. 셋째, 다양한 교과 지식들이 서로 연결할 수 있는 통합에 초점을 두어야 한다. 소프트웨어 자체의 교육 뿐 아니라 타 교과와 연계를 통한 융합 교육을 중요시 하고 있다. <표 II-9>는 인도의 1학년 컴퓨터과학 교육과정의 개념과 사용 기술, 사회적 측면에 대한 내용이다. 이처럼 계열성 혹은 연속성과 계속적인 체계를 통해 학년이나 수준이 올라갈수록 더욱 더 상세해지고 난이도가 높아지는 교육과정을 구성할 필요가 있다는 것을 시사하고 있다.

<표 II-9> 인도 1-2학년 CMC 교육과정

구분	교육 내용
개념	<ul style="list-style-type: none"> <li>• 컴퓨터의 용도가 다르다는 것을 인식</li> <li>• 컴퓨터의 다른 용도 목록</li> <li>• 페인트와 같은 일부 용도의 응용 프로그램 식별</li> <li>• 컴퓨터의 여러 부분 (예 : CPU, 모니터, 마우스, 키보드)을 나열</li> <li>• 각 부품의 관련 기능 식별 (예 : CPU 기능은 컴퓨터의 다른 모든 부분을 제어)</li> <li>• 파일이 무엇인지 설명</li> </ul>
사용 기술	<ul style="list-style-type: none"> <li>• 키보드(화살표, 키 입력) 및 마우스(단일 / 더블 클릭)의 제어 기능</li> <li>• 응용 프로그램 열기, 닫기 및 종료</li> <li>• 간단한 응용 프로그램 (예 : 페인트)을 사용</li> <li>• 파일 열기, 편집 및 저장</li> <li>• 데스크 탑 및 창의 요소 식별 및 조작</li> </ul>
사회적 측면	<ul style="list-style-type: none"> <li>• 컴퓨터를 사용하는 동안 올바른 자세 유지하고 지시 따르기</li> <li>• 컴퓨터를 깨끗하게 유지하기 위한 지침을 따르기 (예 : 컴퓨터 근처에서 음식을 먹거나 마시지 않음)</li> <li>• 공통 자원을 공정하게 공유하기 (예 : 키보드를 사용하는 동안 회전하는 등)</li> </ul>

#### (4) 싱가포르

싱가포르 정보통신개발청(Infocomm Development Authority, IDA)은 2014년 4월부터 코딩 교육 전파 계획인 ‘CODE@SG’를 시작했다. CODE@SG는 보다 어린 나이의 아이들에게 코딩에 대한 관심을 유발해 코딩을 통한 컴퓨팅 사고력을 길러 향후 기술 및 소프트웨어 전문가로 성장할 수 있는 기반을 마련하는 것을 목표로 하고 있다(KOTRA, 2015).

싱가포르는 아시아 최초로 코딩 교육을 도입한 국가이며 취학 전 4세부터 6세를 대상으로 한 ICT 교육 기본 계획이 수립되어 있다. 최신 기술을 활용해 복잡한 도시 문제를 해결하고 통합된 국가의 디지털 전략을 추진하는 스마트네이션(Smart Nation)을 이끌어 나갈 스마트 인재 양성이 목표이다. 스마트네이션으로 거듭나기 위해 해야 할 일 중 하나로 교육을 들었다. 코딩 교육 활성화를 통해 다양한 효과를 기대하고 있고 국가의 인적 역량 강화를 위해 IT 전문가로 성장할 수 있는 기회를 마련하는 데 코딩 교육이 중요한 역할을 수행해 줄 것이라고 기대하고 있다.

<표 II-10>의 싱가포르 코딩 교육과정을 살펴보면 취학 전 어린이들을 대상으로 하는 플레이메이커에는 서킷 스티커, 비봇, 키보, 리틀비츠 등이 있다. 암기나 반복을 통한 훈련이 아니라 직접 도구를 만지고 실험하면서 즐겁게 익히는 체험형, 놀이형 코딩 교육을 통해 자신감과 상상력, 창의력 발달이 교육의 목적이다(MOE. Singapore, 2012). ICT를 활용한 교수·학습의 3가지 지침을 살펴보면 ICT 사용은 아이들의 학습 경험을 보완하고 발달에 적절해야 하며 교사들이 촉진하고 지도해야 하고, 안전을 위해 ICT 활용을 신중하게 검토해야 한다고 제시하고 있다(MOE. Singapore, 2017). 교육 내용과 더불어 교사 지침까지 제공해주고 있어 교육을 위한 지원을 아끼지 않고 있다. 어린 나이의 유아에게 소프트웨어 교육이 의미가 있으며 변화하는 시대를 살아가기 위한 아이들의 미래를 위해서라도 시작의 시기가 중요하다는 것을 시사하고 있다.

<표 II-10> 싱가포르 코딩 교육과정

학년	교육 내용
교수·학습 지침	<ul style="list-style-type: none"> <li>• ICT 사용은 아이들의 학습 경험을 보완하고 발달에 적절해야 한다.</li> <li>• 교사들이 함께 촉진하고 지도해야 한다.</li> <li>• 안전을 위해 ICT 활용을 신중하게 검토해야 한다</li> </ul>
교육 내용	<ul style="list-style-type: none"> <li>• 이야기에서 시작하는 코딩 교육(예 : 책을 통한 스토리를 통해 시작하는 교육)</li> <li>• 비봇을 통해 영어와 수학, 미술 교육</li> <li>• 로봇을 얼마나 움직여야 원하는 곳으로 가게 하는지 생각을 통한 코딩의 원리와 개념</li> </ul>
프로그래밍 언어	<ul style="list-style-type: none"> <li>• 서킷 스티커, 비봇, 키보, 리틀비츠</li> </ul>

4) 해외 소프트웨어 교육의 시사점

현재 코딩이 교육과정에 어떻게 통합되어 있으며 어떤 형태로 교육을 하고 있는지 살펴보고 더욱 발전시키기 위한 노력의 일환으로 2014년 10월 European Schoolnet(2015)에서 발간된 보고서에 따르면 국가수준 교육과정에서 프로그래밍과 코딩을 다루는 나라들은 조사에 포함된 유럽의 21개 국가 중 18개국에 이른다. 전 학년에 걸쳐 국가 교육과정에 코딩이 포함되어 있고 필수나 선택으로 교육하고 있는 국가가 대부분이지만 독립교과로 운영하고 있는 초등학교 과정만 살펴보았다.

<표 II-11> 해외(유럽) 초등학교 코딩교육 현황의 내용을 자세히 살펴보면 초등학교에서 코딩이 필수인 나라는 핀란드, 슬로바키아, 영국이고 선택인 국가는 벨기에, 에스토니아, 프랑스, 이스라엘, 스페인으로 조사되었다. 그러나 필수로 도입을 준비 중이거나 통합교과로 운영 중이지만 독립교과나 핵심교과로 분리 운영할 계획인 국가들도 있었다. 이처럼 국가수준에서 코딩교육을 하는 국가들이 대부분이며 초등학교에서 선택과 필수로 운영하고 있지만 그렇지 않은 국가에서는 학년이 올라갈수록 선택과 필수로 운영하는 국가가 많았다. 이러한 국가들도 코딩 교육 시작 학년을 순차적으로 점점 낮출 것이라고 발표하고 있다.

<표 II-11> 해외(유럽) 초등학교 코딩교육 현황

	초등학교 코딩교육	국가수준	지역사회	시작년도
AUSTRIA	·	○	·	·
BELGIUM(NL)	선택	·	○	·
BULGARIA	·	○	·	·
CZECH REPUBLIC	·	·	·	·
DENMARK	·	○	·	2014
ESTONIA	선택	○	·	·
FINLAND	필수	○	○	2016
FRANCE	선택	○	·	2016
IRELAND	·	○	·	2014
ISRAEL	선택	○	·	1976
HUNGARY	·	○	·	1995
LITHUANIA	·	○	·	1986
MALTA	·	○	·	1997
POLAND	·	○	·	1985
PORTUGAL	·	○	·	2012
SPAIN	선택	○	○	2015
SLOVAKIA	필수	○	·	1990
UK(ENGLAND)	필수	○	○	2014

코딩 교육의 목적은 컴퓨팅 사고력이라는 큰 틀이지만 <표 II-12>의 해외(유럽) 코딩 교육과 교육과정 현황을 자세히 살펴보면 코딩 교육을 통해 논리적 사고력, 문제해결력, ICT 역량, 코딩 기술역량 뿐만 아니라 ICT 직업역량을 통해 고용률을 높이는 목표까지 포함하고 있다.

이처럼 어린 시절 코딩 경험이 학습능력에 상당한 영향을 미친다는 연구를 통해 많은 국가에서 코딩을 통한 컴퓨팅 사고력 향상 소프트웨어 교육의 중요성을 강조하고 있다. 더불어 교사 교육 지원과 비영리 단체들도 교육에 참여하여 많은 성과를 내고 있다. 이를 통해 우리나라도 더 늦기 전에 만 5세부터 소프트웨어 교육을 시작할 필요성이 있으며 방과후 과정, 사교육이 아니라 모든 유아들이 평등하게 교육을 받을 수 있도록 국가수준의 교육과정을 도입해야 할 필요성이 있다는 걸 시사하고 있다.

<표 II-12> 해외(유럽) 코딩교육과 교육과정 현황

	논리적 사고력	문제 해결력	ICT 역량	코딩스킬 역량	ICT 직업 육성	추가 핵심역량
AUSTRIA	○	○	○	○	○	○
BELGIUM(NL)	·	·	○	·	○	○
BULGARIA	○	○	○	○	·	·
CZECH REPUBLIC	○	○	○	○	○	○
DENMARK	○	○	·	·	·	○
ESTONIA	○	○	○	·	·	○
FINLAND	○	○	·	○	·	·
FRANCE	·	·	○	·	○	○
IRELAND	○	○	○	○	·	○
ISRAEL	○	○	○	○	○	○
HUNGARY	○	○	·	·	·	·
LITHUANIA	○	·	·	○	·	·
MALTA	·	·	○	○	·	·
POLAND	○	○	○	○	○	○
PORTUGAL	○	○	·	·	○	○
SPAIN	○	○	·	○	·	○
SLOVAKIA	○	○	·	·	·	·
UK(ENGLAND)	○	○	○	○	○	·

5) 국내

국외 사례를 보더라도 소프트웨어 교육의 시작은 방과후 과정, 사교육이 아니라 국가 수준의 교육과정 안에서 시작해야 한다. 이를 위해 국가 교육과정과 엮박자를 내지 않기 위해 유아를 위한 국가 교육과정인 누리과정과 추후 초·중등과의 연계를 위하여 초등학교, 중학교 교육과정들을 함께 살펴보았다.

(1) 누리과정

누리과정은 유아를 위한 교육·보육 공통과정이다. 유치원과 어린이집에 공통적, 일반적인 기준을 제시해 적용할 수 있는 공통과정이다(교육부 & 보건복지부, 2015). 신체운동·건강, 의사소통, 사회관계, 예술경험, 자연탐구 5개의 영역을 중심으로 구성하여 유치원/어린이집과 친구, 나와 가족, 우리 동네, 동식물과 자연, 건강과 안전, 생활도구, 교통기관, 우리나라, 세계 여러 나라, 환경과 생활, 봄·여

름·가을·겨울 11개의 생활주제로 이루어져 있다.

누리과정 제정의 특성을 살펴보면 기본 생활 습관과 창의 인성교육을 강조하고 있지만 이를 별도의 영역으로 구분 짓지 않고 누리과정 전 영역에서 강조하도록 하고 있다. 누리과정의 구성은 사람과 사람의 관계를 존중하는 것에서부터 출발하여 서로에 대하여 보고 듣고 느끼고 이해하는 것과 서로를 인정하고 수용하고 배려하고 협력하는 것까지 많은 것을 포함하고 있다. 유아는 독창적이고 독특한 능력을 발휘할 수 있는 창의적인 사람으로 자라게 도와주는 것이 바람직하다. 유아기는 창의성 발달이 가장 왕성한 시기이기 때문에 기초적인 사고 능력을 충실히 습득시켜 이를 토대로 유연성 있고 창의적인 사고 능력을 기르는 데 중점을 두어야 한다. 누리과정은 제 발달 영역을 기본 능력으로 함양하고 창의성을 개발할 수 있도록 구성되어 있다. 초등학교 1~2학년의 창의 인성교육 내용 등과 체계적인 연계성을 고려한 교육과정을 제시하고 있다.

누리과정 5개의 영역 중 자연탐구 영역을 살펴보면 ‘필요한 정보나 자료를 수집한다.’ ‘한 가지 기준으로 분류한 자료를 다른 기준으로 재분류 해 본다.’ ‘그림, 사진, 기호나 숫자를 사용해 그래프로 나타낸다.’ 등의 교육내용을 포함하고 있다. 이러한 내용은 유아교육 관점에서 접근보다는 소프트웨어 교육 내용인 자료 표현, 추상화와 연결해 다양한 방법의 교육적 확장이 가능하며 필요해 보인다.

<표 II-13> 누리과정 생활도구로서의 미디어 주요내용

소주제	주요 내용(상세화)	활동명	활동유형 (영역)	누리과정 주요 관련영역
생활 속의 미디어 알아보기	<ul style="list-style-type: none"> <li>- 우리 생활 주변에서 사용되는 미디어의 종류와 역할에 대하여 안다.</li> <li>- 미디어의 변천과정을 안다.</li> <li>- 생활에 필요한 미디어를 사용해 본다.</li> </ul>	생활 속의 미디어	이야기 나누기	자연탐구, 의사소통
		보물을 찾아라	바깥	자연탐구, 사회관계
		우리 집 도우미	동시	자연탐구, 의사소통
		미디어센터 돌아보기	현장 체험	자연탐구, 사회관계
		나도 사진작가	과학	자연탐구
		어떤 이야기를 만들까?	언어	의사소통, 예술경험

소주제	주요 내용(상세화)	활동명	활동유형 (영역)	누리과정 주요 관련영역
미디어가 생활에 주는 유익함 알기	<ul style="list-style-type: none"> <li>- 미디어가 우리 생활에 주는 편리함과 도움에 대하여 안다.</li> <li>- 미디어를 통해 다양한 세계를 접하고 즐긴다.</li> <li>- 미디어의 소중함을 느낀다.</li> </ul>	내게 필요한 정보를 찾아요	과학	자연탐구
		컴퓨터도 예방 주사를 맞아요	이야기 나누기	자연탐구, 사회관계, 의사소통
		미디어 예술가	동화	사회관계, 의사소통
		나도 예술가	미술	자연탐구, 예술경험
		미디어 친구	동시	자연탐구, 의사소통
미디어 유용하게 활용하기	<ul style="list-style-type: none"> <li>- 다양한 미디어의 사용방법을 안다.</li> <li>- 다양한 목적을 위해 미디어를 활용한다.</li> <li>- 미디어 활용을 즐긴다.</li> <li>- 새로운 정보 검색을 위해 미디어를 활용한다.</li> </ul>	컴퓨터 나라에 유치원이 있어요	이야기 나누기	자연탐구, 사회관계
		우리 집 소식을 알려요	미술	사회관계, 예술경험
		컴퓨터 준비 운동	신체	신체운동건강
		우리들이 만든 동화	동극	자연탐구, 예술경험
		영화관 놀이	역할	신체운동건강, 예술경험, 의사소통
		인터넷 나라	음악	자연탐구, 예술경험
미디어 바르게 사용하기	<ul style="list-style-type: none"> <li>- 미디어의 올바른 사용법을 안다.</li> <li>- 미디어 활용 시 개인정보는 함부로 유출하지 않아야 함을 알고 지킨다.</li> <li>- 미디어의 유용함과 유해함을 알고 적절하게 사용한다.</li> <li>- 미디어 사용 시 예절을 지킨다.</li> </ul>	도깨비와 컴퓨터	동화	신체운동건강, 의사소통
		미디어 챔피언	게임	신체운동건강, 사회관계
		개인정보는 소중해요	동화	신체운동건강, 의사소통
		바르게 사용해요	음악	자연탐구, 예술경험
		행복한 미디어 세상	수·조작	사회관계

또한, 11개의 생활주제 중 생활도구 파트에는 ‘생활도구로서의 미디어’와 ‘미래의 생활도구’ 교육활동이 있지만 교사 교육이 제대로 되어있지 않아 의미 있는

교육이 되지 않고 있는 실정이다. 소프트웨어 교육이 유아교육 현장에 성공적으로 자리 잡기 위해서는 교사들의 관심과 수용의지가 우선적으로 고려되어야 할 필요성이 있다(정지현, 2017). <표 II-13>은 생활도구로서의 미디어 활동 부분의 주요 내용이다. ‘우리 생활 주변에서 사용되는 미디어의 종류와 역할’, ‘생활에 필요한 미디어를 사용’, ‘미디어 활용 시 개인정보의 중요성’, ‘미디어 사용 시 예절 지키기’ 등으로 구성되어 있다. 이 부분은 정보윤리 교육, 정보보호 등과 연결해 관련 교육이 가능할 것으로 보인다.

<표 II-14>는 미래의 생활도구 활동 부분의 주요내용이다. ‘로봇이 되어보기’, ‘로봇 디자이너’, ‘미래의 바깥 놀이터’ 등으로 구성된 활동을 생활도구로서의 미디어와 더불어 소프트웨어 교육인 프로그래밍, 피지컬 컴퓨팅과 충분히 연결시킬 수 있는 내용으로 판단된다. 이러한 내용들을 종합해 보면 방과후 과정이나 사교육이 아니라 정규 교과과정인 누리과정 내 흡수하여 충분히 소프트웨어 교육이 가능할 것으로 보인다.

<표 II-14> 누리과정 미래의 생활도구 주요내용

소주제	주요 내용(상세화)	활동명	활동유형(영역)	누리과정 주요 관련영역
새로운 생활도구 알아보기	<ul style="list-style-type: none"> <li>- 새로운 생활도구에 관심 가진다.</li> <li>- 로봇의 종류와 활용에 대해 안다.</li> <li>- 로봇에 대해 친근감을 가진다.</li> </ul>	친환경 정수기	과학	자연탐구, 의사소통
		로봇 되어보기	신체	신체운동건강, 자연탐구
		로봇 디자이너	미술	자연탐구, 예술경험
		미래의 바깥 놀이터	바깥	자연탐구, 예술경험

## (2) 초등 소프트웨어 교육과정

2015 개정 교육과정이 발표되면서 초등학교는 2019년부터 5~6학년 실과 교과에 포함시켜 17시간 소프트웨어 교육을 필수로 실시하고 있다. 그러나 시수가 부족하여 17시간 내에 다룰 수 있는 부분이 상당히 제한적이다. 또한 초등학교 소프트웨어 교육의 핵심역량은 (기술적)문제해결능력, 비판적 사고력, 의사결정능력, 창의력, 기술시스템설계능력, 기술 활용능력이라고 볼 수 있고 중학교 정보

교과에서 추구하는 교과 역량은 정보문화소양, 컴퓨팅 사고력, 협력적 문제해결력(안성훈 & 이상현, 2019)으로 교육과정마저 연계가 되지 않고 있는 실정이다. 현재 필수교과이며 독립교과로 운영되고 있는 학교현장의 교육과정을 살펴보면 내용체계에서 초등과 중등이 함께 표시되지 않은 유일한 교과가 ‘정보’교과라고 할 수 있다(박관우 & 신승기, 2019). [그림 II-1]은 초·중고등학교 2015 개정 교육과정의 정보교과 개편안과 주요 내용이다. 초등학교는 문제해결과정과 프로그래밍을 체험할 수 있도록 하고 정보윤리 의식을 함양할 수 있도록 개편되었다. 중학교는 컴퓨팅 사고력 기반의 문제해결과 간단한 알고리즘, 프로그래밍을 개발할 수 있는 내용으로 개편되었다. 고등학교는 중학교에서 배운 내용들을 토대로 다양한 분야와 융합하여 알고리즘, 프로그램을 설계할 수 있도록 개편되었다.

구분	현행	개편안	주요개편 내용
초등학교 (2019~)	실과 內 ICT 단원(12시간)	실과 內 SW기초교육 (17시간 이상)	<ul style="list-style-type: none"> <li>문제해결과정, 프로그래밍 체험</li> <li>정보윤리의식 함양</li> </ul>
중학교 (2018~)	‘정보’ (선택교과)	‘정보’ 34시간 이상 (필수교과)	<ul style="list-style-type: none"> <li>컴퓨팅사고 기반 문제해결</li> <li>간단한 알고리즘, 프로그래밍 개발</li> </ul>
고등학교 (2018~)	‘정보’ (심화선택과목)	‘정보’ (일반선택과목)	<ul style="list-style-type: none"> <li>다양한 분야와 융합하여 알고리즘, 프로그램 설계</li> </ul>

[그림 II-1] 2015 개정 교육과정(초·중고등학교 정보교과)

실과교과에 관련내용으로 편성되어 있더라도 정보교과에서 초등학교의 정보영역에 대한 내용체계를 통합하여 초등과 중등을 아우르는 위계를 제시하는 것이 바람직하다고 할 수 있으나, 중학교와 고등학교의 내용요소에 대해서만 내용체계로 구성되어 있어 학교급간의 위계가 전혀 고려되지 않은 교육과정으로 편성되었다고 볼 수 있다(박관우 & 신승기, 2019). 초등학교 실과에서 소프트웨어 교육

부분의 성취 기준은 <표 II-15>와 같다(교육부, 2015a). 차기 교육과정에서 이를 위한 심도 있는 논의가 필요하며 초등의 독립교과 편성과 충분한 시수 확보와 더불어 중등과 연계가 진행될 경우를 대비해 유아 소프트웨어 교육도 선제적으로 준비할 필요성이 있다.

<표 II-15> 실과 소프트웨어 영역 세부 내용

영역	세부 내용
학습요소	<ul style="list-style-type: none"> <li>• 소프트웨어가 생활에 미치는 영향</li> <li>• 기초 프로그래밍</li> <li>• 절차적 사고</li> <li>• 사칙 연산·순차·선택·반복 명령</li> <li>• 프로그램 설계와 제작</li> </ul>
성취기준	<ul style="list-style-type: none"> <li>• 소프트웨어가 적용된 사례를 찾아보고 우리생활에 미치는 영향을 이해한다.</li> <li>• 절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.</li> <li>• 프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다.</li> <li>• 자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.</li> <li>• 문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.</li> </ul>

2016년 한국정보교육학회에서 제시한 ‘초등 소프트웨어 교육과정 표준 모델 연구’는 초등과 중등의 연계성을 가지고 컴퓨팅 사고력 뿐 아니라 컴퓨터 과학, ICT 소양, 정보 윤리에 대한 내용이 전체적으로 잘 제시 되어 있다. 영역을 살펴보면 소프트웨어, 컴퓨팅 시스템, 융합 활동 3가지로 구분하여 내용체계와 성취기준을 제시하고 있다. K-6학년까지 위계성과 중학교 교육과정과의 연계성에 따라 내용체계와 성취기준이 잘 정리되어 있어 교육과정으로서의 갖는 의미가 높다고 할 수 있다(김철 외, 2016). 만 5세에 해당하는 빨강 단계의 내용체계를 살펴보면 <표 II-16>과 같다. 국내에서는 만 5세 소프트웨어 교육과정을 처음으로 제시하고 있으나 연령에 해당하는 교육 목적, 세부 내용, 교육 프로그램 및 활동에 대한 예시 등은 많이 부족하거나 없는 편이다.

<표 II-16> 초등 소프트웨어 교육과정 표준 모델 내용체계

영역	핵심개념	△(빨강단계 : 만 5세)
소프트웨어	정보	- 정보의 개념 - 문제해결과정에서의 정보
	문제해결	- 문제해결 경험 - 문제의 의미
	알고리즘	- 일의 순서에 따라 그림이나 문장 나열순서 - 활동 순서에 따라 표현
	프로그래밍	- 물체의 이동과 회전 - 간단한 도형 표현
컴퓨터 시스템	정보기기	- 생활 속 정보기기 - 정보기기의 역할
	운영체제	- 물건 관리 방법 - 일의 관리와 순서
	네트워크	- 물물 교환 - 생활 정보 교환
융합 활동	정보윤리	- 정보기기의 사용자세 - 정보기기 켜고 끄기
	창작도구	- 그림 그리기 - 그림 편집
	로봇	- 로봇의 조작 - 로봇 활용 자세

2015년 교육부에서 발표한 소프트웨어 교육 운영의 기본 방향은 기존의 정보통신기술 교육에서 수행하였던 ICT 소양 및 활용 교육의 관점을 확장하여, 학습자들이 미래 사회에서 살아가는데 필요한 컴퓨팅 사고력을 기반으로 문제를 해결하는 역량을 기르는 것이라고 제시하였다(교육부, 2015b). 초·중등학교에서 이루어지는 소프트웨어 교육은 프로그램 개발 역량보다는 정보윤리 의식과 태도를 바탕으로 실생활의 문제를 컴퓨팅 사고로 해결할 수 있도록 하는 것에 역점을 두고 있다. 초등학교에서는 체험과 활동 위주의 교육, 중학교에서는 개념 이해를 바탕으로 하는 교육, 고등학교에서는 개발과 융합을 통한 소프트웨어 교육을 통해 컴퓨팅 사고력을 가진 창의·융합 인재 육성을 시도하고 있다. [그림 II-2]는 소프트웨어 교육을 통해 추구하는 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고능력을 가진 창의·융합 인재상에 대한 내용이다.

## 컴퓨팅 사고력을 가진 창의·융합 인재



[그림 II-2] 소프트웨어 교육을 통한 인재상

### 6) 국내 소프트웨어 교육의 시사점

교육부에서는 과도한 선행학습으로부터 아이들을 보호하기 위해 유치원 내 코딩교육을 금지하고 있다. 그러나 보건복지부 산하 어린이집에서는 별다른 제재 없이 코딩교육을 실시하고 있어 동일 대상이지만 관계 부처에 따라 다르게 운영되고 있다. 2017년 말 기준 어린이집 유아의 수가 145만 명으로 유치원의 70만 명보다 2배 이상 많아 교육부의 조치에 한계가 있다(유아정책연구소, 2018). 유아 소프트웨어 교육에 대한 학부모들의 관심은 점점 증가하고 있고 이를 대변하듯 사교육 시장을 중심으로 점점 입지를 넓혀가고 있다. 그러나 제대로 된 교육과정 없이 가격이 비싸고 해외와 국내를 막론하고 검증되지 않은 교구를 부문별하게 도입하여 코딩 프로그램 활용 교육으로 점점 변질되어 가고 있다.

국내의 유아대상 소프트웨어 교육 프로그램은 일부 연구자들에 의해 결과가 발표되고 있으며 2017년부터 유아 소프트웨어 교육과 사고력 향상에 관심과 연구들이 꾸준히 증가하고 있는 추세이다. 그러나 교구 기반 소프트웨어 교육 중심으로 이루어져 있거나 소프트웨어 교육 전문성이 다소 부족해 컴퓨팅 사고력 향상이라는 소프트웨어 교육의 궁극적 목적과 연관성이 떨어지는 등 미흡한 실정으로 평가되고 있다. 국내 유아 소프트웨어 교육 선행 연구의 내용들을 살펴보면 <표 II-17>과 같다.

<표 II-17> 유아 소프트웨어 교육 선행 연구

연구자 (발행연도)	연구내용	
노운서 (2010)	제목	만 5세를 위한 컴퓨터 관련 교육활동 개발 및 그 효과에 관한 연구 (학회지 논문)
	교육활동	컴퓨터에 관한 지식, 기능에 관련된 지식과 사용법을 프로젝트 접근법으로 교육
	검사도구	컴퓨터 게임중독 인지도 검사
정봄마지 (2013)	제목	UCC 제작활동이 유아의 디지털 리터러시와 정보윤리의식에 미치는 영향 (석사학위 논문)
	교육활동	디지털 기술 활용에 관한 철학적 원칙과 활용방안에 근거하여 총 7주간의 UCC 제작활동을 연구자가 개발 후 교육
	검사도구	디지털 리터러시 수준 검사, 정보윤리 의식 검사
서영상 (2017)	제목	유아 창의력 향상을 위한 색채블록 코딩 시스템 (박사학위 논문)
	교육활동	미로 탈출 게임은 놀이를 통한 접근의 방법으로 유아들에게 익숙한 아이템과, 색채블록을 이용하여 구상하기, 조작하기, 코딩하기, 모색하기의 과정을 통한 융합 창의 교육으로서의 코딩 교육
	검사도구	검사 미 실시
임명숙 (2017)	제목	스마트 로봇을 활용한 소프트웨어 교육이 유아의 창의성 향상에 미치는 효과 (석사학위 논문)
	교육활동	유아용 스마트로봇 프로그램을 누리교육과정에 따라 창의성 요소를 적용하고, 원 사정에 맞게 프로그램을 재구성하여 교육
	검사도구	유아 통합 창의성 검사(K-ICT)
이우리 (2017)	제목	스마트로봇 알버트BT를 활용한 코딩교육이 만 5세 유아의 의사소통능력 및 창의성에 미치는 효과 (석사학위 논문)
	교육활동	(주)코딩애플레이가 개발한 스마트로봇 알버트BT를 활용한 유아용 코딩교육 프로그램을 연구의 목적에 맞게 12회기로 재구성하여 교육
	검사도구	의사소통능력 검사, 유아 통합 창의성 검사(K-ICT)

연구자 (발행연도)	연구내용	
정민경 (2017)	제목	언플러그드 컴퓨팅을 활용한 STEAM 활동이 유아의 창의성 및 문제해결력에 미치는 효과 (석사학위 논문)
	교육활동	국내외 언플러그드 컴퓨팅 관련 서적과 논문, STEAM 관련 서적과 논문을 참고하여 만 5세에 적절한 언플러그드 컴퓨팅을 활용한 STEAM 활동 총 14개 선정 후 교육
	검사도구	유아 종합 창의성 검사(K-CCTYC), 문제해결력 검사
조준오 홍광표 (2017)	제목	누리과정 기반 유아 소프트웨어 교육과정 구성 방향 탐색 (학회지 논문)
	교육활동	만 5세 유아 소프트웨어 교육을 누리과정과 연계하여 교육 내용을 제시
	검사도구	검사 미 실시
김상연 김상희 (2018)	제목	유아의 컴퓨팅 사고력을 위한 로봇활용교육 프로그램 개발 및 효과 (학회지 논문)
	교육활동	'로보로보' 코딩로봇을 이용한 로봇활용 교육
	검사도구	수학적-논리적 지식 검사, 창의적 문제해결력 검사
박남수 (2018)	제목	게임기반 유아 소프트웨어 교육에서 컴퓨팅 사고력에 영향을 미치는 요인규명 (박사학위 논문)
	교육활동	'키즈 블록' 게임을 기반으로 하는 소프트웨어 교육
	검사도구	창의적 사고력(K-CCTYC) 검사, 논리적 사고력(K-CPM) 검사, 수학적 사고력 검사
김대욱 (2019)	제목	컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 개념과 전략 (학회지 논문)
	교육활동	유아를 위한 언플러그드 코딩에서 필요한 것을 알고리즘 이해하기, 순서도 그리기, 작은 부분으로 나누기, 패턴 찾기, 벌레 잡기, 결과예측하기로 도출
	검사도구	검사 미 실시
원서연 최연철 (2019)	제목	교육용 프로그래밍 언어를 활용한 유아 소프트웨어 교육 경험의 의미 탐색 (학회지 논문)
	교육활동	교육용 프로그래밍 언어 '비스켓'을 사용하여 소프트웨어 교육
	검사도구	검사 미 실시

연구자 (발행연도)	연구내용	
이건우 (2019)	제목	협동적 문제해결학습에 기반한 유아코딩로봇활동이 교사 학습지원 행동과 유아의 수학적 문제해결력에 미치는 영향 (석사학위 논문)
	교육활동	비봇, 알버트, Cubico, Entry, Scratch, Genibo 등을 활용한 활동을 개발하여 교육
	검사도구	교수 행동 관찰 도구, 강원도유아교육진흥원의 유아평가 앱, 뉴질랜드 교육부에서 제공하는 Problem Solving Level 1
정유진 (2019)	제목	유아의 컴퓨팅 사고를 위한 놀이중심 언플러그드 컴퓨팅 프로그램 개발 (학회지 논문)
	교육활동	놀이중심 언플러그드 컴퓨팅 활동을 개발하여 교육
	검사도구	검사 미 실시
한지원 (2019)	제목	프로그래밍 언어를 활용한 유아 소프트웨어교육 실행과정에서 겪은 유아의 경험 (석사학위 논문)
	교육활동	비스킷(Viscuit) 프로그래밍 언어를 활용한 교육
	검사도구	관찰자료 및 면담, 연구일지
황성신 (2019)	제목	소프트웨어 교육 활동 과정에서 나타난 유아의 컴퓨팅 사고 경험 (석사학위 논문)
	교육활동	비스킷(Viscuit) 프로그래밍 언어를 활용한 교육
	검사도구	참여관찰, 연구일지, 면담, 활동 결과물

<표 II-17>에서 보듯이 프로그래밍 언어를 사용한 코딩이나 로봇 활용에 관련된 유아 소프트웨어 교육에 대한 연구들은 매우 다양하였다. 연구자들의 배경을 살펴보면 유아교육 관련 전공자이거나 전문가로 구성되어 있었고 컴퓨터교육 관련 전공자나 전문가의 연구는 없는 실정이다. 선행 연구들을 자세히 분석해 분류해보면 <표 II-18>과 같다. 교육 활동은 코딩과 로봇을 통한 교육이 가장 많았고, 언플러그드, 프로그래밍, 게임 순 이었다. 사용된 검사 도구는 검사를 실시하지 않은 연구가 가장 많았고, 그 외 다양한 사고력 검사, 유아 통합 창의성 검사와 유아 종합 창의성 검사가 뒤를 이었다.

위에서 살펴본 선행 연구의 부족한 점은 단순 활용교육에서 확장하지 못하고 그치는데 있다. 활용교육을 통해 유아들의 호기심을 자극하고 활용 내용을 토대로 유아들이 스스로 더 좋은 방법을 찾아내고 새로운 것으로 연결을 시킬 수 있는 사고력 확장을 통해 컴퓨팅 사고력 향상이 충분히 가능하지만 활용교육으로만 그치는데 문제점이 있다. 이처럼 프로그래밍 언어나 로봇을 선택해 유아 특성에만 중점을 둔 수업 구성으로 교육을 진행할 경우 소프트웨어 교육의 본질에서 벗어나 처음 접하는 소프트웨어 교육이 자칫 활용교육으로 변질될 우려가 있다.

<표 II-18> 유아 소프트웨어 교육 선행 연구 분류

교육 활동		검사 도구	
코딩 및 로봇	5	유아 통합 창의성 검사 유아 종합 창의성 검사	3
언플러그드	3	컴퓨팅 사고력	.
프로그래밍	3	그 외 사고력 검사	2
게임	1	검사 미실시	5
그 외	2	그 외	4

유아 소프트웨어 교육은 놀이와 경험을 통해 문제해결과정에서 알고리즘을 이해하고 프로그래밍의 원리를 자연스럽게 익힐 수 있도록 도와주는 교육이 되어야 한다. 선행 연구에서 살펴본 것과 같이 본 연구에서처럼 교육 목적, 내용체계 개발을 통한 교육 프로그램 개발과 컴퓨팅 사고력 자체에 대한 검사 도구를 활용한 효과성 검증에 대한 실험 연구는 없었다. 유아 통합 창의성 검사(K-ICT)나 유아 종합 창의성 검사(K-CCTYC)등 창의성, 논리성 관련 검사를 실시하거나 하지 않는 연구가 많은 편이다.

창의성이란 새로운 사고를 생산해 내는 능력이고 인간 지적능력의 한 특성이자(Guilford, 1959). 창의적인 생각이란 멀리 떨어진 생각들을 함께 묶는 형식을 띠고 있다고 하였다(Mednick, 1962). 즉, 두 개 이상의 생각들을 새롭게 조합한 결과이다. 창의적인 사고 과정이란 여러 요소를 종합하여 유용한 조합을 새롭게 만들어내는 것이라고 개념화 할 수 있다. Torrance(1978)는 창의적 사고를 결합·

부족한 요인·방해요인 등을 인지하고 이에 관한 가설과 아이디어를 만들어 그 가설을 검증하고, 이를 수정 또는 재검증하여 최종적인 결과를 전달하는 과정이라고 설명하였다. 즉, 기존의 생각이나 관념을 수정, 변화시킴으로써 새로운 생각이나 의견을 포함한 유용한 결과를 도출해 내는 능력이라고 할 수 있다. 반면 컴퓨팅 사고력은 컴퓨터를 활용해 구현할 수 있는 방법으로 문제를 해결하는 접근 방법이다(CSTA, 2017). 컴퓨팅 사고력 구성 요소는 크게 추상화(Abstraction)와 자동화(Automation)를 들 수 있다. 추상화는 ‘어떤 것을 단순화하고 관심을 집중시키기 위해 세부적인 것을 제거하는 과정’이며, ‘그것들의 공통적인 핵심과 본질을 찾기 위해 이를 일반화하는 과정’이다. 추상화는 실제 세계의 문제를 해결 가능한 형태로 표현하기 위한 사고과정이다. 실생활의 다양한 문제를 해결하기 위하여 필요한 자료를 수집하고 분석하여 도표, 그래프 등 필요한 표현 방법을 활용하여 보기 쉽게 나타내고, 복잡한 요소를 작은 단위로 분해하고, 해결에 필요한 변수들을 추출하여 적절한 해결 모델을 설계하는 과정이다(Devlin, 2003; Hazzan, 1999; Kramer, 2007). 즉, 컴퓨터의 문제해결 방식을 이해하고 이를 실생활의 문제해결 과정에 적용하는 능력이라고 정의할 수 있다.

이처럼 창의성과 컴퓨팅 사고력은 서로 다른 영역인 만큼 따로 구분해서 평가하는 것이 바람직하다고 할 수 있다. 컴퓨팅 사고력을 강조하다 보면 창의성은 자연스럽게 좋아질 수밖에 없다. 더불어 소프트웨어 교육 전문가의 관점에서의 유아 소프트웨어 교육 연구와 교육과정 제시 또한 큰 의미를 가진다고 할 수 있다.

### Ⅲ. 유아 소프트웨어 교육과정 개발

#### 1. 유아 소프트웨어 교육의 방향

소프트웨어 교육은 창의적 아이디어를 소프트웨어로 구현하는 사고력 교육을 뜻한다(교육부, 2015b). 유아 소프트웨어 관련 교육 활동의 교육 방향은 ‘소프트웨어에 관한 교육’과 ‘소프트웨어를 통한 교육’ 두 가지를 지향해야 한다(유구종, 1998). 즉, 소프트웨어를 활용할 수 있는 능력도 중요하지만 이를 통해 컴퓨팅 사고력을 함양할 수 있는 활동으로 진행하여야 한다. 더불어, 스마트 기기의 좋은 점과 나쁜 점, 게임에 대한 올바른 인식, 과의존 예방 및 대처 방안, 정보보호 등은 반드시 포함되어야 한다. 또한, 교육기관 내에서만 하는 활동으로 그치지 않고 가정과의 연계 활동으로 이어져 부모와 함께 하는 교육이 되어야 한다.

선행 연구들을 살펴보면 유아교육 전문가들은 소프트웨어 교육에 대해 다양하게 연구 중이지만 소프트웨어 전문가들은 초등과 중등 연구에 많은 시간을 투자하다 보니 아직 시도조차 못하고 있는 실정이다. 유아교육에서는 단순히 코딩 능력, 로봇 활용교육에 초점을 맞춰서 유아 소프트웨어 교육을 확장하고 있다. 그러나 유아 소프트웨어 교육의 핵심은 놀이를 통한 경험을 강조하는 컴퓨팅 사고력 함양과 문제를 더 효과적으로 해결하는 수단인 문제해결력에 기초한 교육이어야 한다. 즉, 코딩, 게임, 로봇 활용 기반으로 연구하고 있는 일반적인 방법을 탈피하여 초등, 중등과 연계할 수 있고 컴퓨팅 사고력 함양이라는 소프트웨어 교육의 궁극적인 목적에 맞는 프로그램 개발이 필요하다.

이에 따라 유아 소프트웨어 교육의 방향은 유아들의 실생활과 학습 주제와 연계한 논리적, 절차적, 체계적 사고력을 반영해야 한다. 동시에 생활 속 문제 해결을 위해 컴퓨터의 사고방식, 즉 문제해결, 코딩(조직화), 알고리즘(논리) 등에 대한 이해와 적용 등이 포함되도록 구안하여야 할 것이다.

유아 소프트웨어 교육은 발달단계에 적합한 놀이를 활용하여 활동이 이루어져

야 한다. 놀이중심 교육은 유아들이 지식을 스스로 구성할 수 있으며 유아의 발달과 요구에 기초한 통합적 경험을 제공할 수 있기 때문이다(나원영, 2019). 다양한 소프트웨어 교육방법 중 유아의 인지발달 단계, 즉 전조작기와 구체적 조작기 초기 단계에서 아이들이 구체물과 상황 및 활동을 통해 소프트웨어 교육이 이루어져야 한다.

#### 1) 유아의 특성 및 교육

우리나라에서 유아란 만 3세부터 만 5세까지 초등학교 취학 전 어린이로 정의하고 있다(교육부, 2018b). 일반적으로 해외에서는 만 5세가 초등학교 입학 연령으로서 이때부터 공교육이 이루어지고 있다(육아정책연구소, 2010). 만 5세의 발달 특성은 가장 짧은 기간 안에 가장 큰 변화와 발달을 경험하는 시기라고 할 수 있다(김현정, 2002). 즉, 발달과 관련된 모든 영역에서 급속한 성장과 변화가 이루어지는 시기이다. 학자들은 유아의 두뇌는 만 6세 전 거의 성숙단계에 이르며, 정신 및 성격형성 등도 만 6세 이후의 성숙기에서보다 급속도로 이루어진다고 하였다. 인지발달을 촉진시키기 위해서 다양한 감각기관들의 자극이 반드시 필요한 시기이다(정옥분, 2002).

따라서 개인의 전 생애에 걸쳐 신체, 인지, 사회, 정서 발달이 짧은 기간 안에 가장 빠르게 일어나며 다른 시기에 비하여 이후의 발달에 미치는 영향력이 매우 크다는 점에서 매우 중요한 시기이다(박수연, 2012). 이 시기는 창의성 발달에 있어 무한한 발전 가능성을 가지고 있기 때문에 창의성 계발의 최적기라고 할 수 있다. 다양한 활동을 제공하고 자발적인 참여를 통한 분위기에서 자연스럽게 교육이 이루어질 수 있도록 구체적인 노력이 필요하다. 이러한 중요한 시기에 해외에서는 국가수준의 교육과정을 통해 만 5세부터 소프트웨어 교육을 진행하고 있으며 관련 지원을 아끼지 않고 있다. 이에 우리나라도 제대로 된 소프트웨어 교육과정과 교육 프로그램을 통해 만 5세 부터 컴퓨팅 사고력 향상이라는 소프트웨어 교육의 궁극적인 목적 달성을 위한 교육을 시작할 필요성이 충분히 있다. 또한 만 5세 소프트웨어 교육의 효과성 검증을 통해 만 3세, 만 4세의 좀 더 어린나이에 조심스러운 교육 확장도 필요하다.

## 2) 유아 소프트웨어 교육의 개념

소프트웨어 교육이란 미래 사회를 살아가기 위해 필요한 문제해결역량을 컴퓨팅 사고력 기반으로 기르는 것을 목적으로 한다. 프로그래밍 교육과 개발 역량에 초점을 맞추기 보다는 정보윤리 의식과 태도를 바탕으로 실생활의 문제를 컴퓨팅 사고로 해결 할 수 있도록 하는 것에 역점을 두는 것이다(교육부, 2015b). 이러한 소프트웨어 교육은 이를 통해 다른 시각으로 세상을 바라보고 생각하도록 하는 것이 가장 중요한 목적 중 하나이다(Helen, 2017).

유아 소프트웨어 교육에 대한 연구는 초기 단계이므로 그 개념을 학자마다 다르게 정의하고 있다. 그러나 시대에서 요구하는 역량의 변화에 따라 소프트웨어 교육을 미래 인재인 유아에게 요구되는 차세대 교육이라는 인식으로 전환되고 있다(Karan, 2006; Papert, 1980). 유아 소프트웨어 교육의 궁극적인 목적 역시 유아들의 컴퓨팅 사고력을 배양할 것을 권면하고 있다(조준오 & 홍광표, 2017). 특히 소프트웨어 교육 자체가 교과 틀에 맞추어진 것이 아니라 일상생활의 문제를 중심으로 이루어져야 한다(김재휘 & 김동호, 2016).

뉴질랜드의 컴퓨터 과학자 Tim Bell(2006) 교수는 언플러그드 활동을 통한 유아 소프트웨어 교육을 제안하였다. 언플러그드 활동은 유아를 포함한 모든 연령과 배경지식이 없는 사람들을 대상으로 컴퓨터과학 활동이 가능하다. 언플러그드 활동은 추상적인 컴퓨터 작동 원리를 컴퓨터 없이 구체적 조작 활동으로 학습자에게 전달하는데, 대표적인 교육내용을 살펴보면 크게 정보의 표현, 알고리즘, 절차 표현, 난해성, 암호화, 컴퓨터와 상호작용영역으로 제시하고 있다(한지원, 2019). 이러한 언플러그드 활동은 구체적인 경험 및 일상생활과 관련된 놀이 형태의 학습을 통해 유아들도 보다 쉽게 접근하고 배울 수 있다는 장점이 있다.

이처럼 소프트웨어 교육은 단순히 지식을 습득하거나, 소프트웨어 프로그램의 기능을 익히는 수준이 아니라 일상생활에서 발생하는 문제를 해결하기 위한 과정을 효과적으로 찾기 위한 사고과정을 다루는 교육이다. 블록기반 프로그래밍 언어인 Scratch Junior를 개발한 Resnick(2007)는 유아 소프트웨어 교육이란 유아가 생각하여 만들고자 하는 구체물을 유아가 사용할 수 있는 교육용 프로그래밍언어를 활용하여 설계 및 창조하는 과정에 참여함으로써 컴퓨팅 사고력을 향상시키는 것을 목표로 한다고 하였다. 국내 연구를 살펴보면 이현철(2016)은 유

아 소프트웨어 교육이란 유아들이 컴퓨팅 사고력 함양을 위하여 컴퓨터과학 분야가 접목된 인지·정의적인 프로그래밍 언어 도구를 활용하는 교육을 의미한다고 하였다. 조준오, 박창현, 홍광표(2017)는 유아들이 교육용 프로그래밍 언어 도구를 활용하여 주변의 생활과 학습 주제 안에서 발생하는 여러 문제들을 해결하기 위한 논리적, 절차적, 체계적인 사고활동으로 정의하고 있다. 또한 소프트웨어 교육은 4차 산업혁명을 대비해 미래 인재인 유아에게 컴퓨팅 사고력 함양을 위해 필요한 교육이라고 정의하고 있다(Clements, 1984; Karen, 2006; Papert, 1987; Resnick, 2008).

이러한 내용들을 종합해보면, 유아 소프트웨어 교육이란 놀이와 경험을 통한 학습으로 생활 속에서 발생하는 다양한 문제들을 효과적으로 해결하는 사고과정을 통해 컴퓨팅 사고력을 길러주는 활동으로 정의할 수 있다. 본 연구에서 제안하는 유아 소프트웨어 교육의 특징은 초·중등과의 연계를 위한 간접적인 형태의 원리 교육과 더불어 유아교사들이 이해하기 쉽고 누리과정 내에 적용할 수 있는 교육과정을 만드는 것이 기준과 목표이다.

### 3) 유아 소프트웨어 교육의 필요성

2015년 세계 경제포럼에서는 현재의 만 7세 이하 유아가 성인이 되어 사회에서 직업을 선택할 때가 되면 65% 이상은 지금 존재하지 않은 직업을 갖게 될 것이라고 예측했다. 이와 같이 4차 산업혁명은 우리가 경험해 보지 못한 것, 지금의 기술과 지식으로 설명이 불가능한 변화일 수 있다. 이러한 변화에 세계 여러 나라에서는 유아 소프트웨어 교육을 강화하고 점점 더 어린 나이부터 교육 시도를 위한 연구를 진행하고 있으며 만 5세부터 정규과정으로 도입하고 있다.

사회 전반적인 변화와 스마트 기기의 발달은 유아의 스마트 기기 사용을 증가시키고 있다. 스마트 기기 이용 연령은 빠르게 하향화되고 사용량은 매우 증가하고 있는 추세이다(이정림 외, 2013). 그러나 제대로 된 정보소양 교육조차 없이 무방비하게 노출되고 있으며 과의존에 대한 우려가 시간이 지날수록 점점 현실로 나타나고 있다. 이러한 부작용을 선제적으로 대비하기 위해 정보윤리 교육, 정보보안 교육, 스마트 기기 과의존 예방 등의 내용이 포함된 교육과정이 필요하다.

현재 초등학교에서는 연간 17시간 이상 소프트웨어 기초소양 교육을 진행하고 있다. 그러나 초등학교 5학년에 소프트웨어 교육을 시작하는 건 다소 늦은 감이 있다. 이러한 현실을 증명이라도 하듯 학부모들은 사교육을 통해 좀 더 어린 나이부터 소프트웨어 교육을 시작하고 있다.

생애 첫 소프트웨어 교육이 부문별한 사교육을 통해 이루어지기 보다는 유아 교육 연구진과 컴퓨터교육 연구진이 함께 노력하여 제대로 된 교육과정과 프로그램을 개발하고 이를 공교육에 적용하여 모두가 교육을 받을 수 있는 시스템이 구축되어야 한다. 사회는 급변하고 있고 변화와 함께 삶의 구조도 다양하고 복잡해지고 있다. 유아들은 미래 사회에 더욱 다양한 해결해야 할 문제들과 직면할 것이다. 이러한 시대에 소프트웨어 교육은 미래 삶을 위해 누구나 배워야 하는 기본소양이기 때문에 보편교육이 되어야 한다. 단순히 암기하고 지식을 습득하는 방법이 아니라 알고리즘과 프로그래밍을 통해 일상생활의 다양한 문제를 발견하고 접근하여 사고하고 최적화된 방법을 찾아 해결할 수 있는 컴퓨팅 사고력을 길러주는 교육이 되어야 한다. 유아들이 소프트웨어 교육이라고 인지하지 않고 놀이, 체험으로 생각할 정도로 자연스럽게 접근할 수 있는 교육이 되어야 할 필요성이 있다.

#### 4) 유아 소프트웨어 교육 교수자

유아 소프트웨어 교육에서 교사 변인은 매우 중요하다(김해림, 2007). 소프트웨어 교육에 대한 유아교사의 인식 및 요구도 조사 결과(조준오 외, 2017)에 따르면 교사들은 필요성에 대해 긍정적으로 인식하고 있고 누리과정과 연계 가능성에 대해서도 ‘보통’ 이상의 긍정적인 응답을 대다수 교사들이 하고 있는 것으로 나타났다. 즉, 소프트웨어 교육의 필요성과 도입에 대해 유아교사들은 비교적 낙관적임을 알 수 있다. 대부분의 유치원 교사들은 유아기 소프트웨어 교육의 필요성을 인식하고 있으나 기초 지식이 부족하고 올바른 유아용 프로그램 선택 방법, 평가 방법에 대한 지식이 없어 시도하지 못하기 때문에 유아를 위한 소프트웨어 교육을 실행하는데 혼란을 경험하고 있다(심숙영, 2003). 이는 교사들이 교실에서 유아들과 컴퓨터로 무엇을 해야 하는지, 어떻게 활용해야 하는가에 대한 통합적 능력이 부족하기 때문이다.

Haugland(2005)는 학급에서 이루어지는 교육과정 목표와 통합되는 소프트웨어와 웹사이트를 선택하는 교사의 능력이 유아의 컴퓨터 경험을 극대화하고 유아가 의미 있는 개념적 지식을 형성할 수 있도록 돕는다고 하였다. 교사가 유아 소프트웨어 교육의 내용 및 방법에 대한 인식 없이 단지 교실에서 컴퓨터를 사용해서 교육하는 것만으로는 소프트웨어 교육의 효과성을 도모하기는 어렵다(Siegle, 2004). 유아와 유아교육 기관의 특성상 유아기 소프트웨어 교육은 컴퓨터실에서 전문 컴퓨터 강사에 의해 이루어지기보다 학급 내 다양한 영역과 통합적으로 학급 담임교사에 의해 이루어지는 것이 가장 적절한 접근이라고 본다(유구종, 1998; Haugland, 2005; Hohmann, 1990; Hyson & Eyman, 1986; NAEYC, 1996).

따라서, 유아 소프트웨어 교육의 교수자 형태는 다음과 같이 유아교사가 전담하는 형태, 유아교사와 소프트웨어 전문 강사가 함께 교육하는 형태, 소프트웨어 전문 강사 전담하는 형태 세 가지로 분류하여 제안할 수 있다.

<표 III-1> 교수자 형태 분류

교수자	방법	수업형태
유아교사	유아교사가 소프트웨어 교육을 전담	정규 수업
유아교사 & 소프트웨어 전문 강사	유아교사와 소프트웨어 전문 강사가 파트별 소프트웨어 교육을 진행	정규 수업+ 방과후 과정
소프트웨어 전문 강사	소프트웨어 전문 강사가 소프트웨어 교육을 전담	방과후 과정

유아교사가 소프트웨어 교육을 전담할 경우 누리과정과 연계한 정규 활동 시간을 활용할 수 있다. 유아교사 즉, 담임교사는 유아들의 특징과 발달형태에 따른 학업 성취도 및 개개인에 대한 성향을 잘 파악하고 있어 수업 중 일어날 수 있는 돌발 상황에 대한 대처를 할 수 있다. 그러나 소프트웨어 전문 지식 없이 단기간 연수를 통해 유아들을 교육하기에는 여러 가지 문제점이 있을 수 있다. 유아교사가 소프트웨어 교육을 전담했을 경우 다음과 같은 장단점이 있다.

첫째, 유아들의 사회적, 심리적 행동발달 및 특성에 맞는 수준별 소프트웨어 교

육이 고려된다.

둘째, 유아 국가 교육과정인 누리과정에 대해 완벽하게 이해하고 있기 때문에 누리과정과 연계할 수 있는 최대한의 효과를 낼 수 있다.

셋째, 유아교사에 따라 소프트웨어 내용에 대한 이해도가 달라 평준화 교육이 어렵고 한정적인 지식으로 인해 상황과 대상별 수준에 따른 교육의 확장이 어렵다.

넷째, 전문지식이 없는 상태로 접근할 경우 유아 소프트웨어 교육의 궁극적 목적 달성이 어렵다.

유아교사와 소프트웨어 전문 강사가 함께 교육을 진행 할 경우 정규 수업과 방과후 과정을 둘 다 활용할 수 있다. 유아교사는 유아들의 특성을 잘 알고 개인에 대해 파악하고 있고, 소프트웨어 전문 강사는 소프트웨어 교육에 대한 내용적 측면에 대해 잘 알고 있어 파트를 나누어 교육을 진행할 경우 효율성을 기대할 수 있다. 그러나 어디까지 유아교사의 영역으로 볼 수 있는지에 대한 문제를 해결해야 한다. 유아교사와 소프트웨어 전문 강사가 함께 소프트웨어 교육을 진행할 경우 다음과 같은 장단점이 있다.

첫째, 유아 교사에게 어려운 파트인 알고리즘 라이프와 플레이 컴퓨팅 분야에 대한 전문 교육이 가능하다.

둘째, 영역별 교수자가 다를 경우 유아들에게 혼란을 줄 수 있고, 교육방식의 차이로 접근이 제약적일 수 있다.

셋째, 소프트웨어 전문 강사는 전문지식을 가지고 있어 체계적이고 통합적인 접근이 가능해 수업 중 유아들의 질문과 호기심을 더욱 확장할 수 있다. 그러나 유아들에 대한 전문지식이 부족하기 때문에 소프트웨어 전문 강사 양성 시 반드시 유아교육, 누리과정에 관한 내용도 함께 교육이 필요하다.

소프트웨어 전문 강사가 소프트웨어 교육을 전담할 경우 방과후 과정을 활용할 수 있다. 소프트웨어 전문 강사가 교육을 진행할 경우 다음과 같은 장단점이 있다.

첫째, 소프트웨어 교육과정을 교육환경과 상황에 맞게 재구성이 가능해 전문지식을 통해 체계적인 교육과 접근이 가능하다.

둘째, 소프트웨어 내용적인 다양한 지식을 이용한 확장성이 가능하고 기술적인

오류에 대한 대처가 빠르다.

셋째, 누리과정에 대한 내용 이해와 더불어 유아의 행동발달과 특성에 맞는 수준별 강의가 고려되어야 한다. 이를 위해 위와 같이 소프트웨어 전문 강사 양성 시 반드시 유아교육, 누리과정에 관한 내용도 함께 교육이 필요하다.

이상에서 제안한 세 가지 형태 중 현실적으로 현재 교육을 시작할 때 가장 효과적인 교수자 형태는 유아교사와 소프트웨어 전문 강사가 함께 교육, 소프트웨어 전문 강사가 교육하는 두 가지 형태를 제안한다. 그러나 추후 국가 교육과정으로 자리 잡고 누리과정 내에 포함하여 교육을 진행할 경우 유아교사가 교육하는 것이 가장 효과적이다. 이를 위해서는 추가적으로 예비 유아교사 양성과정과 유아교사를 위한 소프트웨어 교육 연수와 더불어 유치원 내 소프트웨어 교육을 위한 교실 환경 구성이 선행 되어야 할 것이다.

#### 5) 유아 소프트웨어 교육 교구

소프트웨어 교육에 많은 관심을 가지고 있는 것만큼이나 다양한 교구가 개발되어 판매되고 있다. 이를 교육 현장에 적용하기 위해서는 교구로써 가지는 특징과 교육 환경을 고려하여 선정하는 단계를 거쳐야 한다(이영재 & 김영식, 2016). 바람직한 교구를 선정하는 방법은 주변에서 흔히 볼 수 있는 재료로 활동의 종류에 따라 필요한 교구를 만드는 것이다(오연주 외, 2003). 그러나 소프트웨어 교육의 목적에 맞게 제작이 어려울 경우 구입을 해서 사용을 해야 하는데 이때, 고려해야 할 사항으로는 제작된 교구보다 견고하고 매력적일 것과 반복적이고 지속적인 사용에 필요한 계획을 세워야 하는 것이다(정선희, 2010). 또한, 유아의 발달 수준을 고려해야 한다. 아무리 훌륭하고 매력적인 교구일지라도 그 교구를 사용하게 될 유아의 발달 단계에 적합하지 않다면 좋은 교구라고 할 수 없다.

유아 소프트웨어 교육에서 사용할 수 있는 교구는 크게 언플러그드와 플러그드로 구분할 수 있다. 본 연구에서는 언플러그드 18종과 플러그드 9종에 대해 직접 사용해 보고 특성을 파악해 장점과 단점을 분석하여 시사점을 도출하였다.

<표 III-2> 유아 소프트웨어 교육 교구 분석

구분	교구명	특성	시사점
언플러그드 (총 18종)	스택버거 엔트리봇 I&II 캐치더켓 캐치더독 맛있는 코딩 버그캐치 비밀의 수 코드레이서 코드광 첸토 프로봇 마이크로 로봇 닥터유레카 플레이로봇 I&II 아도큐브 이진수 카드놀이	<ul style="list-style-type: none"> <li>●전체 분석                     <ul style="list-style-type: none"> <li>- 언플러그드 활동을 통해 순차적 사고를 높여주는 보드게임 형태의 교구</li> <li>- 컴퓨팅 사고력 요소인 자료 수집, 자료분석, 자료표현, 문제 분해, 알고리즘/절차에 관한 내용을 언플러그드 활동으로 학습 가능</li> <li>- 대부분 15분 내외의 활동으로 이루어져 있고 2~3인이 함께 할 수 있는 형태(경쟁과 협업 가능)</li> <li>- 규칙을 바꾸어 응용 가능</li> <li>- 사용 연령별 제한이 있음(연령에 맞는 교구를 구입)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>●장점                     <ul style="list-style-type: none"> <li>- 스마트기기 사용이 미숙한 유아를 대상으로 쉽게 수업이 가능</li> <li>- 몸을 움직이고 생각하며 이해할 수 있는 방식으로 지도 가능</li> <li>- 활동지를 사용하여 수업에 적용할 경우 교구비용에 대한 부담이 적음</li> <li>- 생활 속 주제와 연결된 교구가 많아 익숙하게 접근 가능</li> </ul> </li> <li>●제한점 등                     <ul style="list-style-type: none"> <li>- 확장 가능성이 없어 동일한 형태로의 활동만 가능</li> <li>- 혼자서 할 수 있는 활동의 제한</li> <li>- 게임자체에 재미와 경쟁에 집중하다 보면 소프트웨어 교육 목적 상실</li> </ul> </li> </ul>
플러그드 (총 9종)	알버트 큐비코 비봇 대시앤닷 오조봇 뚜루뚜루 큐베토 보틀리 코드앤 고 (로봇마우스)	<ul style="list-style-type: none"> <li>●전체 분석                     <ul style="list-style-type: none"> <li>- 로봇에 대한 호기심을 시작으로 흥미를 유발해 자연스럽게 소프트웨어 교육에 대한 접근가능</li> <li>- 여러 가지 미션을 해결하는 활동을 통해 컴퓨팅 사고력에 대한 접근이 가능</li> <li>- 스마트 기기 연동을 통해 미션 기록으로 활동자료 분석이 가능</li> <li>- 교구 자체 활용 뿐 아니라 스마트 기기와 연동으로 연령별, 단계별 교육으로 확장 가능</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>●장점                     <ul style="list-style-type: none"> <li>- 스택카드를 통해 언플러그드 활동과 플러그드 활동 접목 가능</li> <li>- 로봇만 활용할 수 있는 것이 아니라 스마트 기기와 연동을 통해 여러 가지 주제와 활동으로 확장 가능</li> <li>- 연령별, 단계별 교육 확장이 가능(유아 이외 연령도 사용가능)</li> <li>- 혼자 활동이 가능해 유치원 뿐 아니라 가정에서도 활용 가능</li> </ul> </li> <li>●제한점 등                     <ul style="list-style-type: none"> <li>- 교구의 가격이 비싸 개별적으로 구입이 부담스러움</li> <li>- 소프트웨어 교육보다 로봇 활용교육, 로봇에 대한 흥미 접근으로 교육이 이루어질 수 있음</li> </ul> </li> </ul>

<표 III-2>의 유아 소프트웨어 교구 분석 내용을 자세히 살펴보면 언플러그드 교구는 언플러그드 활동을 통해 순차적 사고를 높여주는 보드게임 형태로 구성

되어 있었고 컴퓨팅 사고력 요소인 자료수집, 자료분석, 자료표현, 문제분해, 알고리즘/절차에 관한 내용을 언플러그드 활동으로 학습 가능했다. 스마트 기기 사용이 미숙한 유아를 대상으로 쉽게 수업이 가능하고 놀이처럼 몸을 움직이고 생각하며 이해할 수 있는 방식으로 수업이 가능하다. 그러나 확장 가능성이 없어 동일한 형태로 활동만 가능하고 2~3명이 필요하기 때문에 혼자서 할 수 있는 활동의 제한과 게임 자체의 재미와 경쟁에 집중하다 보면 소프트웨어 교육 목적을 상실할 수 있는 제한점이 있었다.

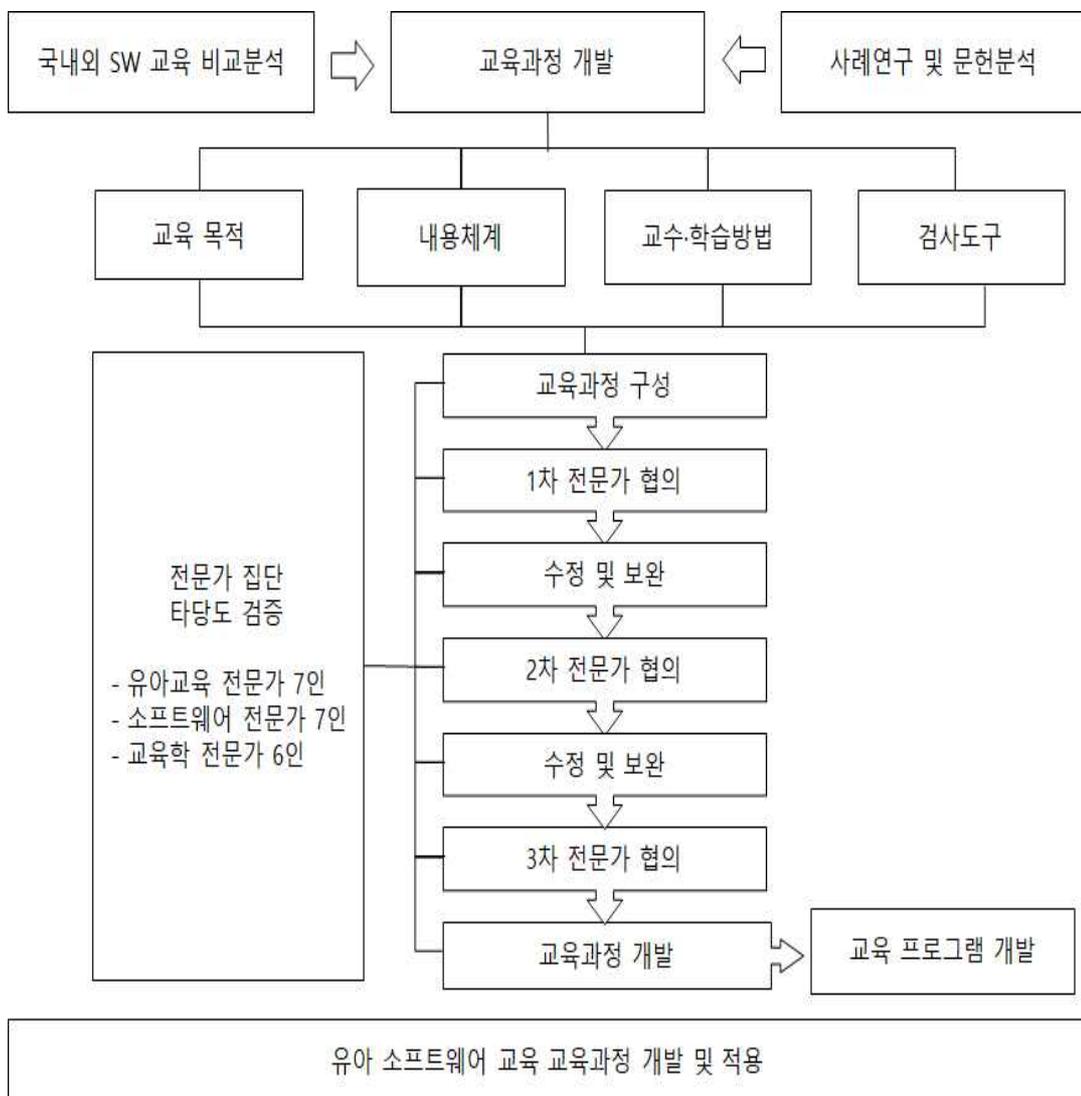
플러그드 교구는 로봇에 대한 호기심을 시작으로 흥미를 유발해 자연스럽게 소프트웨어 교육에 대한 접근이 가능하고 여러 가지 단계별 미션을 해결하는 성취감 있는 활동을 통해 컴퓨팅 사고력을 향상 시킬 수 있다. 로봇 자체로 교육에 활용할 수 있을 뿐 아니라 스마트 기기와 연동이 가능해 연령별, 단계별 교육의 확장이 가능하다. 또한 개별 활동이 가능해 유치원 뿐 아니라 가정에서도 활용 가능하다. 그러나 교구의 가격이 비싸 개별적으로 구입하기 부담스럽고 소프트웨어 교육보다 로봇 활용교육, 로봇에 대한 흥미 접근으로 교육이 이루어질 수 있는 제한점이 있다. 피지컬 컴퓨팅 교육을 위한 교구의 선정 기준을 제시한 선행 연구를 살펴보면 안정성, 교과 호환성, 발달 적절성, 다기능성, 조작의 용이성 및 성능의 신뢰성, 교육과정 적합성, 경제성, 서비스, 내구성 등을 고려할 수 있다고 하였다(이영재 외, 2017).

이와 같은 내용을 종합해보면 유아들에게 놀이와 구체적인 경험을 통한 컴퓨팅 사고력 향상 소프트웨어 교육을 할 경우 컴퓨팅 방법을 이용하여 스스로 쉽게 조작하고 또래 친구들과 함께하는 경험을 통해 협업과 문제해결력을 기를 수 있는 교구를 사용하여야 한다. 활동에 따른 적합한 교구를 사용할 때 유아들이 컴퓨터과학의 기본 개념과 원리를 기반으로 한 컴퓨팅 사고력이 향상되고 새로운 상황에서 사고의 조직화를 통해 문제해결력을 기를 수 있을 것이다. 본 연구에서는 교육 프로그램 활동을 위해 언플러그드 활동 교구는 직접 제작하여 사용하였고 피지컬 컴퓨팅 활동의 교구는 선행 연구에서 제시한 기준에 맞는 S사의 피지컬 컴퓨팅 교구를 사용하였다. 그러나 추후에는 유아 발달 단계와 교육 현장에 적합하고 각 가정과 연결하여 활용이 가능하면서 소프트웨어 교육의 각 차시별 내용 및 목적에 맞는 교구 연구와 제작도 필요하다.

## 2. 교육과정 개발 절차

본 연구는 유아 소프트웨어 교육을 위한 교육 목적, 내용체계를 제시하고 교육 과정을 개발해 그 효과를 검증하는데 목적이 있다.

교육과정을 개발하기 위해 국내외 교육과정 비교 분석, 이론 및 선행 연구를 비교 분석한 내용을 근거로 소프트웨어 교육의 목적, 내용체계, 교육 프로그램, 검사 도구를 구축하였다.



[그림 III-1] 교육과정 개발 절차

본 연구에서 개발한 유아 소프트웨어 교육을 위한 교육과정 개발 절차를 도식화하면 [그림 III-1]과 같다. 제시된 내용을 자세히 살펴보면 교육과정은 교육 목적, 내용체계, 교수·학습 방법, 검사 도구로 구성되어 있다. 교육과정 개발을 위해 국내외 소프트웨어 교육과정 비교분석, 문헌연구, 사례조사, 이론 및 선행 연구를 비교 분석하였으며, 소프트웨어 교육 목적, 내용체계, 교수·학습 방법, 평가 방법 및 검사 도구 선정을 위한 기본 틀을 구성하였다.

대학 교수, 현장 전문가, 현직 교사로 구성된 유아교육 전문가 7인, 소프트웨어 전문가 7인, 교육학 전문가 6인 총 20명으로 이루어진 전문가 집단의 3차에 걸친 타당화 과정을 통해 유아 소프트웨어 교육의 필요성과 목적을 확인하였다. 이러한 과정을 거쳐 개발된 교육 목적과 내용체계에 따라 제시된 교육 활동 예시를 바탕으로 유아교육 교수와 유아교육 전문가들과 함께 소프트웨어 교육 목적을 달성할 수 있는 30개의 교육 프로그램을 개발하였다. 또한 교육의 효과성 검증을 위해 유아용 소프트웨어 교육 효과성 검사 도구와 컴퓨팅 사고력 검사 도구를 구축하여 연구를 완성하였다.

### 3. 교육과정 구성

누리과정 운영은 교육목표와 교육계획 수립, 교수·학습 방법, 교육 평가로 구분할 수 있다(박창현, 2012). 교육목표를 교사가 정확하게 인지할수록 교육의 효과가 높아지게 된다. 교육계획은 학습목표와 관련하여 유아의 발달 특성상 관심과 흥미를 끌 수 있는 구체적이고 체험 가능한 것이어야 한다. 교과서가 없는 유아교육의 특성상 교재, 교구 및 학습매체, 활동방법 또한 구체적이고 실제적이어야 한다. 유아의 수업활동은 놀이중심의 통합적인 교육활동이 되도록 계획하되 또래와 교사, 주변 환경과 다양한 상호작용을 하도록 하며 환경과 유아의 흥미 변화에 따라 융통성 있게 운영될 수 있도록 대체활동을 미리 계획해 두어야 한다(김영희 & 유경애, 2008). 평가는 교육목표와 내용을 준거로 개인차를 고려하여 실생활에서 관찰 가능하고 유아의 발달을 최적화하는 것을 목적으로 유아평

가가 이루어지도록 해야 한다.

유아 소프트웨어 교육과정 개발 전 교육의 필요성과 기본 방향, 적절한 시작시기와 교육 방법 등에 대한 전문가 요구분석을 실시하였다. 대학 교수, 현장 전문가, 현직 교사로 구성된 소프트웨어 교육 전문가 3인, 유아 교육 전문가 3인의 인터뷰를 시행하였다. 이 과정을 통해 얻어진 주요 의견을 살펴보면 다음과 같다.

소프트웨어 중심사회로 변화함에 있어 문제해결능력, 논리적 사고 향상을 위해 유아 소프트웨어 교육은 필요하다. 초등과의 연계성을 위해서라도 저학년으로 확대되어야 하고 좀 더 어린 나이부터 시작할 필요성이 있다. 놀이나 게임 위주의 논리적 사고력을 향상시킬 수 있는 형태가 필요하지만 교육이 즐거운 놀이 형태를 취하더라도 소프트웨어가 게임이라는 오류를 범하지 않도록 해야 한다. 누리과정의 전반적인 교육과정 내에서 놀이중심 활동, 언플러그드 활동, 유아 수준의 컴퓨팅 사고력 향상과 문제해결력 향상을 위한 소프트웨어 교육 방향이 필요하다.

누리과정 혹은 만 5세부터 소프트웨어와 친숙해지는 교육을 시작할 필요성이 있고 놀이를 통해 체험하고 생각할 수 있는 교육은 4~5세를 포함하여 유치원 교육과정에서 가능하다. 소프트웨어를 활용한 활동이 유아 주도적으로 이루어지도록 하고, 조사나 탐구활동, 문제해결 과정 등을 포함한다는 점에서 만 5세가 적절하다. 미디어 및 소프트웨어를 다루는 시기가 점차 빨라지고 있으므로 발달에 맞춰 유아기에 이루어져야 한다. 4차 산업혁명 시대를 살아갈 유아들이 사회적 환경에 적응할 수 있도록 능력과 태도를 유아기부터 길러줄 필요가 있고 시대적 변화와 요구를 고려해볼 때 일정 부분 미디어 리터러시를 포함한 소프트웨어 교육이 필요하다.

유아의 생활과 통합되어 제시되어야 하고 유아의 발달수준에 적합해야 한다. 미디어를 통해 접하는 정보에 대해 비판적으로 사고하여 활용하고, 미디어가 우리 삶에 주는 긍정적, 부정적 측면을 이해하며, 미디어를 대하고 활용하는 올바른 태도에 중점을 두어야 한다.

누리과정의 영역들과 연계되어 각 생활주제에서 지속적으로 다루어져야 하고 초등교육과 연계를 살펴 수준이 설정되어야 한다. 유아교육의 기본적인 생활원

리, 놀이중심, 유아중심 등 교수학습 원리에 따라 이루어져야 한다. 소프트웨어 교육을 위한 소프트웨어 교육이 아닌 유아교육이 지향하는 교육에 부합하고 유아에게 보다 구체적이고 다양한 경험을 준다는 차원에서 소프트웨어가 활용되는 것이 바람직하다.

전문가들의 의견을 통해 시사점을 도출하면 소프트웨어 중심사회로의 변화와 4차 산업혁명 시대를 살아갈 유아들의 문제해결능력, 논리적 사고를 통해 사회적 환경에 적응할 수 있도록 고려할 때 소프트웨어 교육은 필요하다고 판단된다. 시작 시기는 만 5세가 적당하고 놀이중심, 유아중심, 생활중심의 활동을 고려해야 하고 누리과정 영역들과 초등교육과의 연계를 살펴 수준을 설정해야 한다. 더불어 미디어의 노출이 시작된 유아들의 정보윤리 교육과 관련 내용의 교육도 포함할 필요성이 있다. 이러한 내용들을 고려하여 교육과정을 구성하였다.

#### 1) 교육 목적

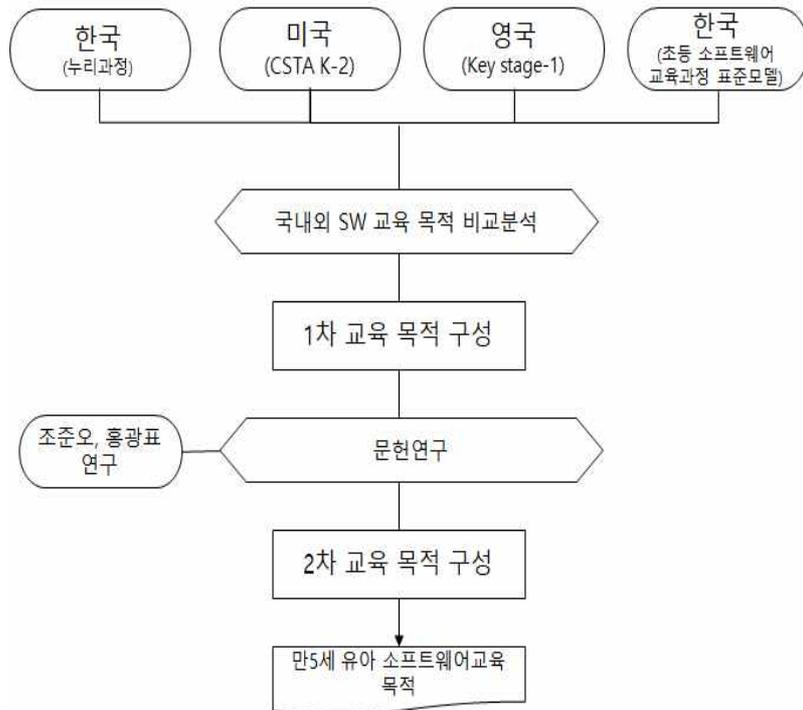
Tyler(1949)는 교육과정을 계획할 때 먼저 달성하고자 하는 교육의 목적을 설정하고 이러한 목적을 달성하기 위해 필요한 학습내용의 선정, 학습활동의 효과적인 조직, 학습활동의 효율적인 평가 과정을 포함해야 한다고 하였다.

유아 소프트웨어 교육은 4차 산업혁명 시대에 소프트웨어 중심의 미래인재 역량 개발을 통해 미래 사회를 살아갈 구성원으로서 필요한 능력과 태도를 주도적인 학습 경험과 놀이중심의 통합 과정을 통해 기르는 것이다. 이러한 유아의 소프트웨어 교육을 위해 먼저 교육의 목적을 설정하였다.

##### (1) 교육 목적 도출

교육 목적은 미국 K-12의 1단계, 영국 컴퓨팅의 Key Stage-1, 우리나라의 경우 누리과정, 초등 소프트웨어 교육과정 표준 모델들을 비교 분석 하였다. 추후 누리과정 및 초중등과 연계되어 이루어질 수 있는 목표라고 할 수 있다.

교육 목적 도출 절차를 살펴보면 [그림 III-2]과 같다. 1차 교육 목적 도출 후 조준오, 홍광표의 '누리과정 기반 유아 소프트웨어 교육과정 구성 방향 탐색' 연구에서 제시한 내용을 2차로 비교 분석하였다.



[그림 III-2] 교육 목적 개발 절차 모형도

각각에서 제시하고 있는 교육의 목적을 분류나 구분을 없애고 공통적인 내용으로 묶어 재배열 해 보았다. 교육 목적 분류를 자세히 살펴보면 <표 III-3>과 같다. 공통되는 요소를 묶고 불필요한 요소를 제거하는 과정을 거쳐나갔다.

<표 III-3> 교육 목적 분류표

누리과정	미국	영국	초등 소프트웨어 교육과정 표준모델	조준오, 흥광표 연구
<ul style="list-style-type: none"> <li>일상생활에서 수학적 과학적으로 이해하고 해결하기 위한 문제 해결 능력을 기르는 것</li> </ul>	<ul style="list-style-type: none"> <li>계산적 사고는 문제를 해결하고 시스템을 디자인하고 새로운 지식을 만들며 현대 사회에서 컴퓨팅의 사용에 대한 지식을 높여줄 것으로 기대</li> </ul>	<ul style="list-style-type: none"> <li>컴퓨팅 용어로 문제를 분석할 수 있으며, 그 문제들을 해결하기 위해 컴퓨터 프로그램을 작성하는 실질적인 경험을 충분히 반복</li> </ul>	<ul style="list-style-type: none"> <li>컴퓨터 과학의 기초 지식을 기반으로 새로운 발상을 보편 타당화</li> <li>창의적, 논리적, 비판적이고, 계산적인 사고능력을 기르는 것</li> </ul>	<ul style="list-style-type: none"> <li>컴퓨팅 사고력 기를 수 있도록 구성(생활 속 문제를 해결하는 역량과 연결)</li> </ul>

누리과정	미국	영국	초등 소프트웨어 교육과정 표준모델	조준오, 홍광표 연구
<ul style="list-style-type: none"> <li>• 자신과 함께 생활하는 사람들과 사회적 관계를 맺고 더불어 생활할 수 있는 태도</li> </ul>	<ul style="list-style-type: none"> <li>• 교사 및 가족, 동료로부터 도움을 받아 전자적으로 정보를 수집하고 의사소통</li> <li>• 기술을 사용하여 동료, 선생님, 다른 사람과 협업적, 협력적으로 일할 수 있음</li> </ul>	<ul style="list-style-type: none"> <li>• 정보 기술을 분석적으로 평가하고 적용하여 문제를 해결 (수업 내용을 살펴보면 의사소통/협업을 강조)</li> </ul>	<ul style="list-style-type: none"> <li>• 새로운 문제들의 탐구 및 해결 능력을 갖출 수 있도록 문제 해결력, 탐구력, 의사결정 능력을 기르는 것</li> </ul>	
<ul style="list-style-type: none"> <li>• 미디어의 유용함과 유해함을 인식하여 다양한 목적에 따라 효율적으로 활용할 수 있는 능력함양</li> <li>• 전자미디어의 무절제한 사용에 대한 피해를 막고 유아가 예절을 지켜 미디어를 사용할 수 있도록 인성교육과 연계</li> </ul>	<ul style="list-style-type: none"> <li>• 디지털 시민의 책임(법적, 윤리적 행동을 실천)</li> <li>• 테크놀로지 사용에 대한 긍정적인 사회적, 윤리적 행동을 알아낼 수 있음</li> </ul>	<ul style="list-style-type: none"> <li>• 정보 통신 기술에 대해 책임감 있고 유능하며 자신감 있고 창의적인 사용자</li> </ul>	<ul style="list-style-type: none"> <li>• 지식 정보를 효과적으로 활용하여 다원적인 가치를 창조하고 세계인과 공유할 수 있는 시민이 될 수 있도록 정보윤리, 정보 활용 능력, 의사소통 능력을 기르는 것</li> </ul>	<ul style="list-style-type: none"> <li>• 정보윤리의식과 인성 교육에 중점 (알고리즘, 프로그래밍, 놀이중심 언플러그드 활동 등과 융·복합적으로 제공)</li> </ul>

미국 K-12의 1단계, 영국 컴퓨팅의 Key Stage-1, 우리나라의 경우 만 5세 누리과정, 초등 소프트웨어 교육과정 표준 모델, 조준오, 홍광표 연구의 교육 목적을 분류 후 재정리한 결과는 <표 III-4>와 같다. 각 교육 목적은 컴퓨팅 사고력, 정보문화 소양, 협력적 해결역량을 공통적으로 제시하고 있었으며, 추가적으로 5세 누리과정에서는 의사소통 능력을 영국, 미국에서는 프로그램 능력을 교육의 목적으로 제시하고 있었다. 조준오, 홍광표 연구에서는 컴퓨팅 사고력과 정보문화 소양을 강조하고 있었으나 협력적 해결역량과 추가적인 능력에 대한 언급은 없었다. 우리나라 국가 교육과정인 중학교 정보의 교육 목적과 맥락을 같이 하는 내용이어서 중학교와의 연계도 충분히 가능할 것으로 확인되었다.

<표 III-4> 교육 목적 분류 정리표

교육 목적	5세 누리과정	미국	영국	초등 소프트웨어 교육과정 표준모델	조준오 흥광표
컴퓨팅 사고력	△	○	○	○	○
정보문화 소양	△ (생활도구 파트)	○	○	○	○
협력적 해결역량	○	○	△ (수업 내용에 포함)	○	X
의사소통 능력	○	X	X	○	X
프로그램 작성 능력	X	△ (수업 내용에 포함)	○	X	X

이와 같이 국내외 교육의 목적과 선행 연구를 비교분석 한 결과 크게 세 가지 컴퓨팅 사고력(Computational Thinking), 정보문화 소양(Information Cultural Literacy), 협력적 해결역량(Collaboration)를 강조하고 있었다. 이러한 근거로 도출된 유아 소프트웨어 교육의 목적과 구체적인 설명에 대하여 살펴보면 다음과 같다.

첫째, 컴퓨팅 사고력은 컴퓨터 과학의 기본 개념과 원리를 이용하여 실생활의 다양한 문제를 창의적이고 효과적으로 해결할 수 있는 능력이다.

둘째, 정보문화 소양은 인성교육과 연계를 통해 디지털 정보 사회에서 윤리적으로 행동하고 정보기기를 목적에 맞게 활용할 수 있는 능력이다.

셋째, 협력적 해결역량은 효과적인 표현과 의사소통을 통해 함께 정보를 수집하고 공유해 창의적으로 문제를 해결 할 수 있는 능력이다.

컴퓨팅 사고력 함양을 유아 소프트웨어 교육의 첫 번째로 목적으로 제안한 이유는 소프트웨어 교육의 가장 중심이 되고 본질적인 목표이기 때문이다.

(2) 타당도 검증

국내외 교육과정 비교분석 및 문헌연구를 근거로 교육 목적을 구성하였다. 타당성 검증을 위하여 3차에 걸쳐 적합성 검증 연구가 수행되었다.

구성된 교육 목적은 대학 교수, 현장 전문가, 현직 교사로 구성된 유아교육 전문가 7인, 소프트웨어 전문가 7인, 교육학 전문가 6인이 타당성 검토를 시행하였다. 전문가들은 각 교육 목적별 의견, 설명에 대한 의견과 목적 전체에 대한 전

반적인 의견 등을 제시하였다.

이 과정을 통해 얻어진 교육 목적에 대한 전문가들의 주요 의견들을 살펴보면 다음과 같다.

첫째, 컴퓨팅 사고력에 대한 전문가 의견은 컴퓨터과학의 개념과 원리 등을 이용해 문제를 해결하는 것 보다는 ‘일상생활과 주변 사물, 자연 환경 등을 컴퓨터 과학적으로 생각하는 능력과 태도’를 목표로 두고 이를 위해 다양한 경험을 할 수 있도록 해야 한다. 즉, 누리 교육과정에서 추구하고자 하는 수학적, 과학적 생각 능력의 연장선에서 본 교육의 목표를 생각하는 것이 바람직하다. 컴퓨팅 사고력은 새로운 디지털 시대가 요구하는 기본 소양으로서, 우리나라 소프트웨어 교육의 핵심 가치 목표이기 때문에 타당하다.

둘째, 정보문화 소양 함양에 대한 전문가 의견은 기술을 익히기 전에 바르게 사용할 수 있도록 윤리교육을 하는 것은 안전교육처럼 반드시 필요한 과정이다. 다만, ‘정보문화 소양 함양’이라는 목표가 이러한 설명을 잘 표현할 수 있고 유아에게 알기 쉽게 누리 교육과정에서 사용하는 용어들을 참고하여 적절한 수준으로 재 서술할 필요성이 있다. 정보기기 과다 노출, 유아의 기초학습 능력 증진을 저해하는 비속어, 은어(예: 유튜브 게임 해설 방송 등)의 남용에 취약한 상황에서 유아의 바람직한 태도 함양을 지원하기 위한 내용도 필요하다.

셋째, 협력적 해결역량에 대한 전문가 의견은 유아 수준에서는 문제 해결보다 자신의 생각을 표현해보는 활동에 집중할 필요가 있고, 협업은 문제 해결보다는 배려와 존중과 같은 개념으로 정보문화 등에서 다루는 것도 방법으로 고려할 필요가 있다. 문제를 해결할 수 있는 능력보다 협력적인 능력을 배양하는 것에 중점을 두어야 한다. 다양하게 문제를 해결할 수 있다는 창의적인 사고, 협력적인 문제해결을 위한 효과적인 표현 및 의사소통 능력을 배양해야 한다.

전문가 의견을 통한 시사점을 도출해보면 컴퓨팅 사고력은 새로운 디지털 시대가 요구하는 기본 소양으로서, 우리나라 소프트웨어교육의 핵심 가치 목표이다. 정보기기 사용 과다 노출과 흔히 접할 수 있는 영상의 비속어, 은어에 대한 내용에 대해 심각하게 받아들이고 있었고, 자신의 생각을 표현해보는 활동 등을 강조하고 있었다. 협의된 전문가 의견 결과를 수렴하고 수정, 보완하여 교육의 목적을 완성하였다.

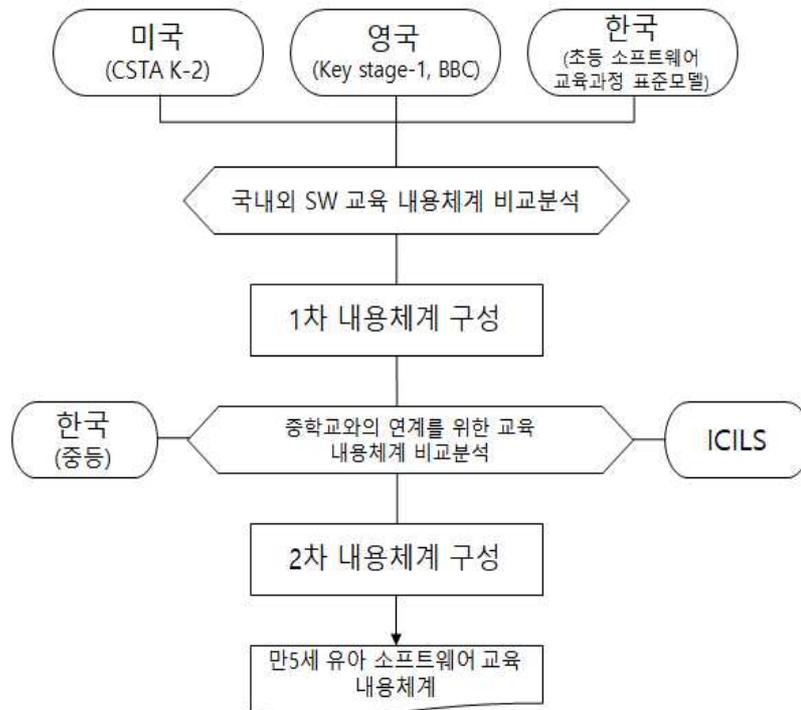
## 2) 내용체계

내용체계는 도출된 교육 목적을 달성할 수 있는 내용들을 위주로 충분히 담아낼 수 있도록 고려되었다.

### (1) 내용체계 도출

내용체계 개발을 위해 미국 K-12의 1단계, 영국 Key Stage-1, 영국 BBC 컴퓨팅 교육, 영국 컴퓨팅 교과서, 한국의 초등 소프트웨어 교육과정 표준 모델 내용체계를 비교 분석 하였다.

내용체계 개발 절차를 살펴보면 [그림 III-3]과 같다. 1차 내용체계 구성 후 중학교와의 연계를 고려하여 중학교 정보 교육과정의 내용체계와 국제컴퓨터소양연구(ICILS, 2013)의 내용체계를 2차로 비교 분석하였다.



[그림 III-3] 내용체계 개발 절차 모형도

각각의 단원에서 내용들을 꺼내 기존의 분류나 구분을 없애고 비슷한 항목끼리 나열해 공통되는 요소로 묶어 재배열 해 보았다. 내용체계 분류를 자세히 살펴보면 <표 III-5>와 같다. 공통되는 요소를 묶고 불필요한 요소를 제거하는 과정을 거쳤나갔다.

<표 III-5> 내용체계 분류표(미국, 영국, 한국 초등)

미국(K-2)	영국(Key Stage-1)	초등 소프트웨어 교육과정 표준 모델
<ul style="list-style-type: none"> <li>• 디바이스                             <ul style="list-style-type: none"> <li>- 적절한 소프트웨어 선택 활용</li> <li>- 기술에 대한 요구 사항과 선호도가 다름을 인식</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 작품 만들기</li> <li>• 비디오 만들기</li> <li>• 애니메이션이 만들어지는 과정</li> </ul>	<ul style="list-style-type: none"> <li>• 그림그리기</li> <li>• 그림편집</li> </ul>
<ul style="list-style-type: none"> <li>• 하드웨어와 소프트웨어</li> <li>• 문제해결(trouble shooting)                             <ul style="list-style-type: none"> <li>- 하드웨어와 소프트웨어 문제 상황 설명</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 하드웨어</li> <li>• 소프트웨어</li> </ul>	<ul style="list-style-type: none"> <li>• 정보기기 켜고 끄기</li> </ul>
<ul style="list-style-type: none"> <li>• 저장                             <ul style="list-style-type: none"> <li>- 정보를 저장, 검색, 인출, 수정 및 삭제, 데이터로서 정보 정의</li> </ul> </li> <li>• 수집, 시각화 및 변환(transformation)</li> <li>• 추론 및 모델(Interface &amp; Models)                             <ul style="list-style-type: none"> <li>- 표, 그래프등 시각화 된 데이터 속 패턴 식별, 예측</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 목적에 따라 디지털 콘텐츠를 생성, 분류, 저장, 처리, 불러오는 기술을 사용</li> </ul>	<ul style="list-style-type: none"> <li>• 정보의 개념</li> </ul>
<ul style="list-style-type: none"> <li>• 변수(variables)                             <ul style="list-style-type: none"> <li>- 정보를 표현하는 숫자나 다른 기호를 이용하여 프로그램이 어떻게 데이터를 저장하고 조작하는지를 모델링</li> </ul> </li> <li>• 모듈화(Modularity)                             <ul style="list-style-type: none"> <li>- 문제를 해결하기 위해 필요한 절차를 정교한 명령 순서로 분해</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 논리적으로 추론하여 간단한 프로그램의 동작을 예측 (프로그래밍과 연계)</li> <li>• 이미지 분류(이미지에 예, 아니오 질문하기와 답하기)</li> </ul>	<ul style="list-style-type: none"> <li>• 문제해결 과정에서의 정보</li> <li>• 문제 해결 경험</li> <li>• 문제 의미</li> </ul>
<ul style="list-style-type: none"> <li>• 알고리즘                             <ul style="list-style-type: none"> <li>- 작업을 완료하기 위한 알고리즘을 작성하고 수행, 일상의 과정을 모델링</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 알고리즘의 정의</li> <li>• 알고리즘이 디지털 기기에 프로그램으로 어떻게 적용되는지 이해</li> </ul>	<ul style="list-style-type: none"> <li>• 일의 순서에 따라 그림이나 문장 나열</li> <li>• 활동 순서에 따라 표현</li> </ul>
<ul style="list-style-type: none"> <li>• 프로그램 개발                             <ul style="list-style-type: none"> <li>- 프로그램의 예상결과 설명하는 계획 작성</li> <li>- 오류 디버깅</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 프로그램은 분명하고 명확한 명령에 따라 실행</li> <li>• 간단한 프로그램 만들고 오류를 찾아내 제거</li> </ul>	<ul style="list-style-type: none"> <li>• 물체의 이동과 회전</li> <li>• 간단한 도형 표현</li> </ul>
<ul style="list-style-type: none"> <li>• Control                             <ul style="list-style-type: none"> <li>- 문제 해결을 위한 순차구조와 반복 구조 활용한 프로그램 개발</li> </ul> </li> </ul>		
<ul style="list-style-type: none"> <li>• 문화                             <ul style="list-style-type: none"> <li>- 새로운 컴퓨팅 기술과 우리 삶의 비교</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 학교 밖에서 일반적으로 정보통신 기술이 어떻게 활용되고 있는지 인지</li> </ul>	<ul style="list-style-type: none"> <li>• 생활 속 정보기기</li> </ul>

미국(K-2)	영국(Key Stage-1)	초등 소프트웨어 교육과정 표준 모델
<ul style="list-style-type: none"> <li>• 사회적 상호작용</li> <li>• 안전성, 법과 윤리</li> <li>• 사이버 보안</li> </ul>	<ul style="list-style-type: none"> <li>• 개인 정보를 보호하면서 안전하고 책임감 있게 기술을 사용</li> <li>• 인터넷이나 기타 온라인 플랫폼에 접속이나 콘텐츠에 이상이 있을 때 어디에 도움을 받아야 할지 인지</li> </ul>	<ul style="list-style-type: none"> <li>• 정보기기의 사용자세</li> </ul>
	<ul style="list-style-type: none"> <li>• 로봇을 활용한 프로그래밍 (BBC)</li> </ul>	<ul style="list-style-type: none"> <li>• 로봇의 조작</li> <li>• 로봇 활용 자세</li> </ul>
	<ul style="list-style-type: none"> <li>• 협동능력 기르기</li> <li>• 과제 수행 결과를 토론하고 개선</li> <li>• 조별 활동을 통해 협동 능력 함양</li> <li>• 정보통신기술 활용에 관해 토론</li> </ul>	

2차 내용체계 구성을 위해 중학교와 연계를 고려해 현재 필수교과로 운영하고 있는 유일한 국가 교육과정인 중학교 정보 교육과정의 내용체계와 빠진 내용이거나 추가적으로 중요하게 다루어야 할 내용 요소를 찾기 위해 국제컴퓨터소양연구(ICILS)의 내용체계를 추가적으로 비교 분석하였다. <표 III-6>의 내용체계 분류표를 자세하게 살펴보면 중학교 정보 교과의 내용체계에는 협업에 대한 내용이 포함되어 있지 않았고 ICILS에서 제시하고 있는 내용체계에는 프로그래밍, 프로그램 개발, 피지컬 컴퓨팅에 관련된 부분이 포함되어 있지 않았다. 그러나 그 부분을 제외한 나머지 부분은 전체적인 맥락을 같이하고 있는 것을 확인할 수 있었다.

<표 III-6> 내용체계 분류표(한국 중학교, ICILS)

한국(중학교)	ICILS
· 컴퓨팅 기기의 구성과 동작 원리	· 컴퓨터 시스템에 대한 지식과 이해 · 컴퓨터 사용에 대한 지식과 이해
· 자료의 유형과 디지털 표현 · 자료의 수집	· 정보 접근 및 평가 · 정보 관리 · 자료수집 및 표현
· 정보의 구조화	· 정보 변환 · 정보 생성
· 문제 이해 · 핵심요소 추출 · 알고리즘 이해 · 알고리즘 표현 · 입력과 출력 · 변수와 연산 · 제어 구조 · 프로그래밍 응용	· 문제분석 및 형식화  · 알고리즘  · 프로그램
· 정보사회의 특성과 진로	· 정보공유 - 디지털 의사소통
· 개인정보와 저작권 보호 · 사이버 윤리	· 정보를 책임 있고 안전하게 사용
· 센서 기반 프로그램 구현	

미국 K-12의 1단계, 영국 Key Stage-1, 영국 BBC 컴퓨팅 교육, 영국의 컴퓨팅 교과서, 한국의 초등 소프트웨어 교육과정 표준 모델에서 내용체계를 분류 후 정리한 결과는 <표 III-7>과 같다.

각 내용체계들을 정보윤리, 정보기기, 협업, 자료와 정보, 추상화, 알고리즘, 프로그래밍, 프로그램 개발, 피지컬 컴퓨팅으로 <표 III-7>과 같이 재분류 할 수 있다. 협업은 미국과 영국의 경우 내용체계에는 없고 교육과정 내 흡수하여 교육하고 있지만, 유아 소프트웨어 교육의 성격과 목적으로 정의하고 있는 중요한 요소인 만큼 내용체계에 추가하였다. 피지컬 컴퓨팅은 미국, 영국에서는 내용체계에는 없지만 교육과정 내 흡수하여 교육하고 있고, 한국 초등 교육과정에는 없는 내용이지만 중학교와의 연계 및 ICILS의 내용체계에서도 제시하고 있는 중요한 요소인 만큼 유아 소프트웨어 내용체계에 추가 하였다. 프로그래밍은 영국에서는 교육과정 내 흡수하여 교육하고 있고 한국 초등 교육과정에는 없는 내용이지만

이 역시 현재 초등학교 실과 교육과정에 포함하고 있는 내용으로 중요한 요소인 만큼 내용체계에 추가하였다.

<표 III-7> 내용체계 분류 정리표

내용	미국	영국	한국(초등)
정보윤리	○	○	○
정보기기	○	○	○
협업	△ (교육과정 내 흡수)	△ (교육과정 내 흡수)	X
자료와 정보	○	○	△
추상화	○	○	○
알고리즘	○	○	○
프로그래밍	○	△ (교육과정 내 흡수)	X
프로그램 개발	○	○	○
피지컬 컴퓨팅	△ (교육과정 내 흡수)	△ (교육과정 내 흡수)	○

국내의 교육과정 내용체계 비교·분석을 바탕으로 개발된 내용체계의 내용 및 세부내용과 비교에 제시된 교육 프로그램 예시는 <표 III-8>과 같다. 비교에서 제시한 교육 프로그램 예시는 실생활 또는 교육기관에서 쉽게 경험 가능하면서 유아가 자발적으로 활동에 참여할 수 있고 흥미를 불러일으킬 수 있는 내용과 주제를 통해 문제해결력과 컴퓨팅 사고력을 함양할 수 있도록 고려하여 활동방법을 제시하였다. 이 예시를 참고하여 유아교육 전문가들이 현장에 적용할 수 있는 프로그램으로 개발하였다. 유아들이 실생활에서 쉽게 경험할 수 있는 이해하기 쉬운 활동으로 구성하였다. 추후 제시된 교육과정을 바탕으로 이질적이지 않도록 각 교육기관의 교육적, 지역적, 환경적, 형태적 특성을 고려하여 다양한 활동으로 확장이 가능할 것이다.

<표 III-8> 내용체계 및 교육 프로그램 예시

내용	세부내용	비고
정보윤리	<ul style="list-style-type: none"> <li>- 개인정보 보호</li> <li>- 사회적 상호작용</li> <li>- 안전하고 책임감 있는 온라인 사용</li> </ul>	<ul style="list-style-type: none"> <li>- 나의 정보를 보호하는 방법 알기</li> <li>- 온라인 세상에서 나는 누구인지 알아보기</li> </ul>
정보기기	<ul style="list-style-type: none"> <li>- 정보기기의 활용과 정보문화</li> <li>- 컴퓨터 주요 장치에 대한 이해</li> <li>- 상황에 따라 필요한 소프트웨어 선택해 사용</li> <li>- 하드웨어와 소프트웨어 문제 상황 설명, 해결방법</li> </ul>	<ul style="list-style-type: none"> <li>- 스마트폰과 컴퓨터는 일상생활에 어떻게 활용되는지 알기</li> <li>- 스마트폰 나라에 들어가 보기</li> <li>- 버스 시간표 앱으로 알아보기</li> </ul>
협업	<ul style="list-style-type: none"> <li>- 과제 수행 결과를 토론하고 개선</li> <li>- 조별 활동을 통해 협업하여 문제 해결</li> <li>- 정보공유, 디지털 의사소통</li> </ul>	<ul style="list-style-type: none"> <li>- 친구들과 함께하는 활동에서 나의 역할 찾아보기</li> <li>- 내가 알고 있는 정보를 친구에게 알려주고 함께 이야기 해보기</li> </ul>
자료와 정보	<ul style="list-style-type: none"> <li>- 자료 저장, 검색, 인출, 수정 및 삭제</li> <li>- 목적에 따른 자료 분류 및 시각화</li> </ul>	<ul style="list-style-type: none"> <li>- 온라인에서 필요한 자료를 검색해서 저장해보기</li> <li>- 목적에 맞게 재배치 해보기</li> </ul>
추상화	<ul style="list-style-type: none"> <li>- 문제 의미 파악</li> <li>- 문제 분석 및 분해</li> <li>- 핵심요소 추출</li> </ul>	<ul style="list-style-type: none"> <li>- 집에서 유치원 오는 길에 보이는 건물 순서대로 간소화하여 그려보기</li> <li>- 추상화 그림 카드 사용하기(그려보기)</li> </ul>
알고리즘	<ul style="list-style-type: none"> <li>- 알고리즘의 정의 및 표현</li> <li>- 문제 해결 과정 표현(알고리즘 표현)</li> <li>- 언플러그드 기반 문제 해결 방안</li> </ul>	<ul style="list-style-type: none"> <li>- 아침에 일어나 유치원에 가기위한 준비과정을 알고리즘으로 작성하기</li> </ul>
프로그래밍	<ul style="list-style-type: none"> <li>- 순차구조와 반복구조 등 제어구조</li> <li>- 명령에 따라 실행되는 프로그램 확인</li> <li>- 프로그래밍 도구를 이용한 프로그램 작성</li> </ul>	<ul style="list-style-type: none"> <li>- 프로그래밍 할 수 있는 장난감 사용해 보기</li> </ul>
프로그램 개발	<ul style="list-style-type: none"> <li>- 작성할 프로그램 기획</li> <li>- 프로그램의 예상결과 설명하는 계획 작성</li> <li>- 명령을 순서대로 입력하여 프로그램 구현</li> <li>- 프로그램의 오류를 찾아서 제거</li> </ul>	<ul style="list-style-type: none"> <li>- 프로그래밍 장난감을 움직이는 프로그램 기획하기</li> <li>- 어떻게 움직이는지 친구들에게 설명하기</li> </ul>
피지컬 컴퓨팅	<ul style="list-style-type: none"> <li>- 로봇 켜고 끄기/ 구성요소 연결</li> <li>- 로봇 안전하게 사용하고 부품 정리</li> <li>- 개발한 프로그램 실행</li> </ul>	<ul style="list-style-type: none"> <li>- 프로그래밍 장난감이 어떻게 움직일지 예측하고 실행하기</li> <li>- 프로그램의 오류를 찾아내고 제거해 보기</li> </ul>

(2) 타당도 검증

내용체계도 교육 목적과 동일하게 타당성 검증을 위하여 3차에 걸쳐 적합성

검증 연구가 수행되었다. 전문가들은 각 영역의 핵심개념을 기준으로 도출된 내용 요소의 타당성을 중점으로 검토하고 핵심개념 자체에 대한 의견, 핵심개념 구성에 관한 의견 등을 제시하였다.

이 과정을 통해 얻어진 내용체계에 대한 전문가들의 주요 의견들을 살펴보면 다음과 같다.

첫째, 소프트웨어 자체에 대한 정보를 제공하다보면 실제 아동들이 정보기기에 대한 흥미가 감소될 우려가 있으므로 소프트웨어에 대한 체험 위주의 교육내용을 구성해야 한다. 문제를 해결하고 정보를 나누는 활동만으로 충분한 학습이 가능한데 평가를 하게 되면 좋은 평가를 받고자하는 마음이 생기게 되어 창의적인 활동이 오히려 저해될 수 있다.

둘째, 정보 기기 오남용이 유아들에게 얼마나 안 좋은 결과를 가져다주는지와 정보 기기를 제대로 사용하면 얼마나 편리한지에 대한 양면성을 적절히 조화시키면서 교육해야 하고 특히 정보윤리(네티켓)에 대해 중점을 두는 방안이 필요하다.

셋째, ‘순차구조, 반복구조, 제어구조’가 소프트웨어 교육에 필요하지만 유아 수준에서 이러한 내용과 단어는 어려울 수 있어 쉬운 용어로 풀어서 표현해야 한다. 만 5세 유아에게 적합한 흥미롭고 실제적인 ‘활동’ 위주의 교육 목표 및 내용 제안이 필요하다.

넷째, 유아교육에 있어서 알고리즘에 대한 교육 시 이에 대한 개념에 대한 이해가 병행되는 것이 필요한지 아니면 개념에 대한 이해와 무관하게 활동이나 경험 등을 통해 습득하도록 하는 것이 바람직한 것인지 이론적인 뒷받침이 필요하다. 초등학교 저학년과 연계되는 과정으로 순차적 및 연계성을 고려한 배열이 필요하다. 또한 향후 프로그램 방향성 및 구성 시 만 5세 유아에게 적합한 흥미롭고 실제적인 ‘활동’ 위주의 교육목표 및 내용 제안을 고려해볼 필요가 있다.

다섯째, 유아 대상 교육인 만큼 만 5세 유아기에 꼭 해야 하는 활동인지에 대한 고려가 우선시 되어야 한다. 유아 단계에서 소프트웨어 교육은 놀이여야 한다. 실패와 부정적인 경험은 부작용을 야기할 수 있으므로 놀이중심의 내용이 되도록 구성할 필요가 있다. 또한, 한글은 학교에서 배운다는 현 정책기조에 역행하지 않도록, 프로그래밍 도구는 절대적으로 ‘비언어적’이어야 한다.

전문가 의견을 통한 시사점을 도출해보면 정보기기의 올바른 사용법과 정보윤리에 대한 내용은 반드시 필요하며, 유아 소프트웨어 교육은 놀이처럼 진행되어야 하고 활동 위주의 교육이 이루어져야 한다는 내용들을 강조하고 있었다. 유아기에 꼭 필요한 활동인지에 대한 고려가 우선시 되어야 하며 실패와 부정적인 경험을 쌓지 않도록 놀이중심의 내용으로 구성되어야 한다. 또한 어려운 용어들을 유아가 알기 쉽게 순화해서 사용할 필요성이 있고, 개념에 대한 이해와 무관하게 활동이나 경험 등을 통해 습득하도록 하는 것이 바람직 할 것으로 보인다는 의견도 있었다. 협의된 전문가 의견 결과를 수렴하고 수정·보완하여 이를 바탕으로 내용체계를 완성하였다.

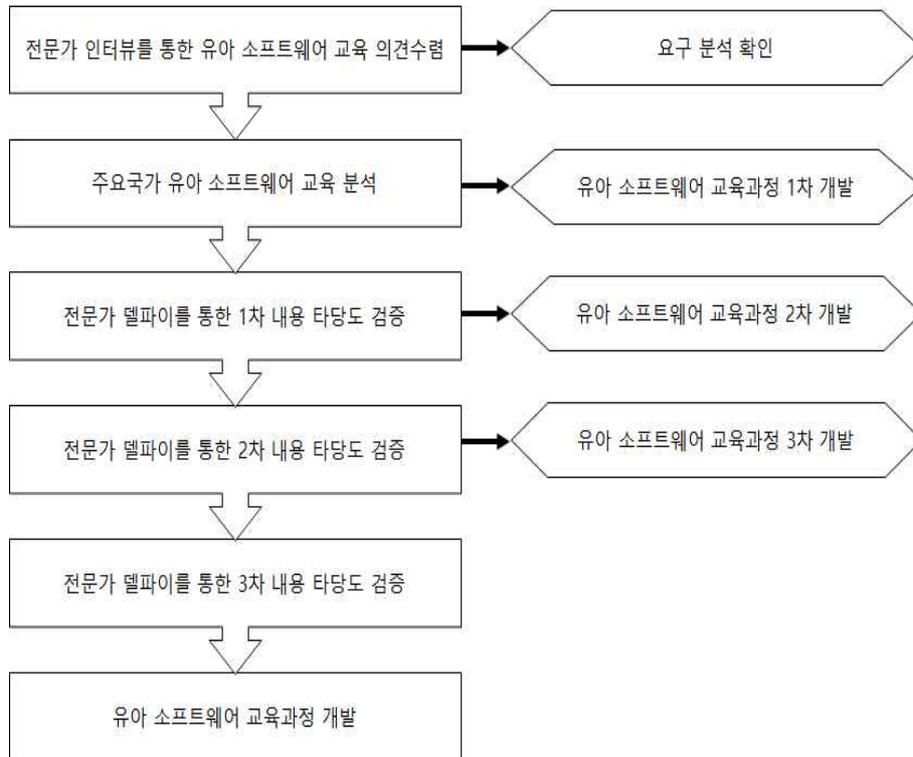
### 3. 교육과정 검증 및 결과

주요 국가 사례분석을 통해 교육 목적을 도출하고 이에 기반 한 내용체계를 수립 하였다. 도출된 내용을 토대로 소프트웨어 전문가, 유아교육 전문가, 교육학 전문가, 20명을 대상으로 3차에 걸쳐 타당도 검증을 위해 델파이를 진행하였다. 교육과정 검증 절차를 살펴보면 [그림 III-4]와 같다.

첫째, 전문가 인터뷰를 통한 유아 소프트웨어 교육에 관한 의견수렴을 통해 요구분석을 확인하였다.

둘째, 국내외 유아 소프트웨어 교육을 비교 분석하고 선행연구, 문헌고찰 과정을 거쳐 1차 교육과정을 개발하였다.

셋째, 개발된 교육과정은 3차에 걸쳐 전문가 델파이를 통해 타당도 검증을 거쳐 수정 및 보완하여 교육과정을 완성하였다.



[그림 III-4] 교육과정 검증 절차

내용 타당도는 Lawshe(1975)가 제시한 내용 타당도 비율(Content Validity Ratio :CVR) 을 바탕으로 분석하였다. <표 III-10>는 교육 목적과 내용 영역에 대한 3차 최종 델파이 내용 타당도(content validity ratio :CVR) 결과 값이다. 내용 타당도 식은 (1)에 의해 산정된다. 여기서,  $n_e$ 는 적합하다고 응답한 사례 수이고 N은 전체 패널 수이다.

$$CVR = \frac{n_e \frac{N}{2}}{\frac{N}{2}} \quad \text{————— (1)}$$

내용 타당도 비율은 패널 수에 따라 최솟값을 제시하고 있으며, 최솟값 이상이 되었을 때 문항에 대한 내용 타당도가 있는 것으로 판단된다(Ziglio, 1996). 본 연구에서는 20명이 설문에 응답하였으며, 그에 따라 평가항목의 타당도는 CVR 최솟값 0.42 이상을 기준으로 하였다. <표 III-9>에서 보는 바와 같이 모든 평가항

목이 CVR 최솟값 기준 0.42 이상으로 나타나 내용타당도가 확보되었다.

<표 III-9> 교육과정 3차 최종 델파이 내용 타당도

		M	SD	M	SD	M	SD	M	SD	CVR	Ma	Q1	Q3			
1. 목표	컴퓨팅 사고력 함양	4.17	0.41	4.57	0.79	4.71	0.49	4.50	0.61	0.90	5.0	4.0	5.0			
	정보문화소양 함양	4.33	0.52	4.71	0.49	4.57	0.53	4.55	0.51	1.00	5.0	4.0	5.0			
	협력적 문제해결력 배양	4.50	0.55	4.71	0.49	4.43	0.79	4.55	0.60	0.90	5.0	4.0	5.0			
2. 내용 영역	1. 디지털 킬러	1-1 정보윤리	1-1-1	4.33	0.52	4.86	0.38	4.66	0.47	4.63	0.48	0.90	5.0	4.0	5.0	
			1-1-2	4.17	0.41	4.86	0.38	4.86	0.38	4.65	0.49	1.00	5.0	4.0	5.0	
			1-1-3	4.50	0.55	5.00	0.00	4.71	0.76	4.75	0.55	0.90	5.0	5.0	5.0	
		1-2 정보기기과 소프트웨어	1-2-1	4.67	0.52	4.86	0.38	4.86	0.38	4.80	0.41	1.00	5.0	5.0	5.0	
			1-2-2	4.00	0.63	4.43	0.79	4.71	0.76	4.40	0.75	0.70	5.0	4.0	5.0	
		1-3 협업	1-3-1	4.33	0.52	4.57	0.79	4.29	0.76	4.40	0.68	0.80	4.5	4.0	5.0	
			1-3-2	4.33	0.82	5.00	0.00	4.71	0.76	4.70	0.66	0.80	5.0	5.0	5.0	
			1-3-3	4.26	0.76	4.86	0.38	4.57	1.13	4.58	0.82	0.70	5.0	4.4	5.0	
		2. 알고리즘 라이프	2-1 자료	2-1-1	4.67	0.52	4.86	0.38	4.71	0.76	4.75	0.55	0.90	5.0	5.0	5.0
	2-1-2			4.00	0.00	4.57	0.53	4.86	0.38	4.50	0.51	1.00	4.5	4.0	5.0	
	2-2 추상화		2-2-1	4.33	0.82	4.86	0.38	4.86	0.38	4.70	0.57	0.90	5.0	4.8	5.0	
			2-2-2	4.00	1.10	4.14	0.69	4.71	0.49	4.30	0.80	0.80	4.0	4.0	5.0	
			2-2-3	4.00	0.63	4.14	0.90	4.71	0.49	4.30	0.73	0.70	4.0	4.0	5.0	
	2-3 알고리즘		2-3-1	4.67	0.52	4.57	0.53	4.86	0.38	4.70	0.47	1.00	5.0	4.0	5.0	
			2-3-2	4.67	0.52	4.57	0.53	4.86	0.38	4.70	0.47	1.00	5.0	4.0	5.0	
			2-3-3	4.50	0.55	4.57	0.53	4.86	0.38	4.65	0.49	1.00	5.0	4.0	5.0	
	2-4 프로그래밍		2-4-1	4.50	0.55	4.57	0.53	4.86	0.38	4.65	0.49	1.00	5.0	4.0	5.0	
			2-4-2	4.17	0.75	3.86	0.90	4.57	0.79	4.20	0.83	0.50	4.0	3.8	5.0	
	3. 플레이 컴퓨팅		3-1 언플러그드/SW 컴퓨팅	3-1-1	4.33	0.52	4.43	0.53	4.57	0.79	4.45	0.60	0.90	4.5	4.0	5.0
				3-1-2	4.33	0.52	4.29	0.76	4.57	0.79	4.40	0.68	0.80	4.5	4.0	5.0
		3-1-3		4.17	0.75	4.29	0.76	4.43	1.13	4.30	0.86	0.70	4.5	4.0	5.0	
		3-2 피지컬 컴퓨팅	3-2-1	4.33	0.52	4.57	0.53	4.71	0.49	4.55	0.51	1.00	5.0	4.0	5.0	
			3-2-2	4.33	0.52	4.14	0.90	4.14	1.46	4.20	1.01	0.70	4.0	4.0	5.0	
			3-2-3	4.33	0.52	4.20	0.69	4.71	0.49	4.42	0.59	0.80	4.2	4.0	5.0	
3-2-4			4.17	0.75	4.14	0.90	4.29	1.11	4.20	0.89	0.60	4.0	4.0	5.0		

1차에서는 도출된 교육 목적 및 내용체계에 대한 타당성 델파이를 진행하였으며 적합도 검정에서 빈도분석 결과 16.97(84.67%)로 나타났다. 교육 목적은 3개가 수정되었으며, 내용 요소 중 14개는 유지하고, 9개가 수정되었으며 2개가 병합, 2개가 삭제되었다. 유아교육과 유아 소프트웨어 교육의 목적 지향점을 함께

고려하여 수정안을 마련하였다. 2차에서는 수정된 교육 목적 및 내용체계 중요도에 대한 타당성 델파이를 진행하였으며 적합도 평균 분석결과 4.3으로 나타났다. 수정안에 대한 적합도 평가 의뢰 결과 평균값 및 사분위 범위 벗어난 응답을 분석하여 4개 요소의 내용을 수정하였다. 3차에서는 교육 목적 및 내용체계 최종 점검을 진행하였으며 적합도 평균은 4.5 CVR값은 .86으로 기준 .42이상으로 나타나 내용 타당도가 확인되었다. 분석결과와 주요 수정 사항을 정리해서 살펴보면 <표 III-10>과 같다.

<표 III-10> 분석결과와 주요 수정 사항

구분(시기)	조사목적	분석결과	주요 수정 사항
1차	도출된 교육목적 및 내용체계 타당성	빈도분석 :16.97(84.67%)	<ul style="list-style-type: none"> <li>유아교육의 목적, 소프트웨어 교육의 지향점을 종합 고려하여 수정안 마련</li> <li>3개 목적 수정, 내용요소 중 유지 14개, 수정 9개, 병합 2개, 삭제 2개 요소</li> </ul>
2차	수정된 교육목적 및 내용체계 중요도	적합도 평균분석: 4.3	<ul style="list-style-type: none"> <li>수정안에 대한 적합도 평가 의뢰</li> <li>평균값 및 사분위 범위 벗어난 응답 분석하여 4개 요소 내용 수정</li> </ul>
3차	교육목적 및 내용체계 최종 점검	적합도 평균: 4.5 CVR: 0.86	<ul style="list-style-type: none"> <li>적합도 평균: 4.5</li> <li>CVR값이 0.86으로 기준 0.42이상으로 나타나 내용타당도 확인됨</li> </ul>

교육과정 개발 연구를 통해 최종 구성된 유아 소프트웨어 교육을 위한 교육 목적은 <표 III-11>과 같다. 교육 목적은 컴퓨팅 사고력 함양, 정보문화소양 함양은 동일하게 제시하고 있지만 협력적 해결역량에서 협력적 문제해결력 배양으로 수정이 이루어졌고 설명에 대해서도 전체적으로 수정, 보완이 이루어졌다.

교육 목적의 구체적인 내용을 살펴보면 다음과 같다.

첫째, 컴퓨팅 사고력 함양은 문제를 추상화하고 해결하는 과정을 놀이와 생활 속에서 자연스럽게 익혀 논리적 사고를 기반으로 새로운 문제 해결의 핵심 능력을 기르는 것이다.

둘째, 정보문화 소양 함양은 인성교육과 연계를 통해 디지털 쉼터에서 윤리적으로 행동하고 모범을 보이며 예절을 지켜 정보기기를 목적에 맞게 활용할 수 있는 사회적 상호작용 능력을 기르는 것이다.

셋째, 협력적 문제해결력 배양은 새로운 문제 해결 활동에 있어 효과적인 표현과 의사소통을 통해 함께 정보를 수집하고 공유해 결과를 도출할 수 있는 협동능력을 기르는 것이다. 이러한 핵심역량은 타 교과 학습에도 기반이 될 뿐 아니라 점차 기술화되고 세계화되어 가는 시대에 필요한 기술들을 갖추는데 핵심적인 요소가 될 것이다.

<표 III-11> 유아 소프트웨어 교육 목적

교육 목적	설명
컴퓨팅 사고력 함양	컴퓨팅의 원리를 이용하여 유아들이 경험하는 일상생활의 다양한 문제를 창의적이고 효과적으로 해결해보는 경험을 한다.
정보문화소양 함양	디지털 정보사회에서의 윤리적 행동을 이해하고, 알맞은 소프트웨어를 활용하는 것이 도움이 됨을 경험한다.
협력적 문제해결력 배양	친구들과 함께 정보를 수집하고 공유하는 과정을 통해 협력적으로 문제를 해결해보는 경험을 한다.

교육과정 개발 연구를 통해 최종 구성된 교육 목적을 달성할 수 있는 유아 소프트웨어 교육 내용체계는 <표 III-12>와 같다. 1-2에는 정보기기에서 정보기기와 소프트웨어로 수정되었으며, 1-3은 자료와 정보에서 자료로 수정되었다. 3-1은 프로그램 개발에서 언플러그드/SW 컴퓨팅으로 수정되었으며 내용 요소들 또한 유아의 발달과 이해하기 쉬운 단어, 소프트웨어 교육의 목적에 맞게 조금씩 수정, 보완되었다.

내용 영역에 대해 자세하게 살펴보면 다음과 같다.

첫째, 디지털 쉼터는 디지털 정보 사회의 구성원으로서 윤리의식을 갖추고 정보기기를 효과적으로 활용하여 문제를 해결할 수 있는 내용을 학습하는 영역이다. 또한 디지털 정보의 공유와 디지털 의사소통 등을 통한 협업으로 보다 효과적이고 창의적인 문제 해결이 가능함을 경험하는 영역이다. 핵심개념으로 ‘정보

윤리’, ‘정보기기’, ‘협업’을 포함한다.

둘째, 알고리즘 라이프는 일상생활의 문제들을 효과적으로 해결하기 위해 필요한 자료와 정보를 찾고, 어렵고 복잡한 문제를 단순화하여 문제 해결 방안을 절차적으로 표현하고 실행하는 것을 학습하고 경험하는 영역이다. 핵심개념으로 ‘자료와 정보’, ‘추상화’, ‘알고리즘’, ‘프로그래밍’을 포함한다.

셋째, 플레이 컴퓨팅은 직접 프로그램을 계획하여 만들고, 실행, 평가하는 것을 늘이처럼 학습하고 경험하는 영역이다. 프로그램의 유형에 따라 핵심개념으로 ‘언플러그드 또는 SW 컴퓨팅’, ‘피지컬 컴퓨팅’을 포함한다.

<표 III-12> 유아 소프트웨어 교육 내용체계

내용 영역	핵심개념	내용 요소
1. 디지털 컬처	1-1. 정보윤리	1-1-1. 개인정보보호의 의미와 중요성 알기 1-1-2. 개인 정보 보호 실천하기 1-1-3. 사이버예절 지키기
	1-2. 정보기기와 소프트웨어	1-2-1. 생활 속 정보기기 경험하기 1-2-2. 나에게 필요한 소프트웨어 선택하고 활용하기
	1-3. 협업	1-3-1. 온/오프라인으로 정보를 공유하기 1-3-2. 친구와 협동하여 문제를 해결하기 1-3-3. 함께 해결한 과정에 대해 평가하기
2. 알고리즘 라이프	2-1. 자료	2-1-1. 필요한 자료 찾아보기 2-1-2. 목적에 따라 자료를 분류하여 표현하기
	2-2. 추상화	2-2-1. 해결하고자하는 문제를 인식하기 2-2-2. 문제를 쉽게 해결하기 위해 작은 문제로 나누기 2-2-3. 문제를 단순하게 표현하기
	2-3. 알고리즘	2-3-1. 문제를 해결하기 위해 단계적으로 생각하기 2-3-2. 문제 해결의 절차를 간단한 그림이나 기호로 표현하기 2-3-3. 절차에 따라 문제를 해결하기
	2-4. 프로그래밍	2-4-1. 프로그래밍 도구 경험하기 2-4-2. 순차구조, 반복구조, 선택구조 경험하기
3. 플레이 컴퓨팅	3-1. 언플러그드 /SW 컴퓨팅	3-1-1. 프로그램을 계획하고 결과 예상하기 3-1-2. 간단한 프로그램 만들기 3-1-3. 프로그램 결과물 평가하기
	3-2. 피지컬 컴퓨팅	3-2-1. 로봇의 사용 방법 및 주의 사항 알기 3-2-2. 로봇에 적용할 프로그램 만들기 3-2-3. 프로그램에 따른 로봇 동작 확인하기 3-2-4. 로봇 프로그램 결과물 평가하기

개발된 내용체계는 교육 속에 컴퓨팅 원리와 개념을 자연스럽게 포함시켜 쉽게 이해할 수 있는 내용으로 구성하였다. 이를 통해 교육의 목적인 컴퓨팅 사고력, 정보문화 소양을 함양하고 협력적 문제해결을 배양 할 수 있을 것이다.

## IV. 교육 프로그램 적용 및 효과 분석

### 1. 연구 방법

본 연구에서 개발한 유아 소프트웨어 교육을 위한 교육과정 및 교육 프로그램을 적용 후 효과를 검증하기 위하여 만 5세 유아를 대상으로 2주간 총 10차시의 수업을 진행하였다. 이에 따라 유아의 소프트웨어 교육 효과성 검사와 컴퓨팅 사고력 검사를 실시하였다.

#### 1) 연구 대상

본 연구는 병설유치원 만 5세 유아 방과후 과정 원생 중 학부모가 연구에 동의하는 유아를 대상으로 실시하였다. 연구자가 보낸 연구 설명서와 동의서에 서면 동의한 2개 반 유아 41명을 선정하였다. 담임들과 사전면담을 통해 연구 대상 유아들은 모두 소프트웨어 교육을 받은 적이 없는 것으로 확인했다. 각 반마다 소프트웨어 전문 교사 1인과 담임교사 1인이 함께 수업을 진행하였다. 본 프로그램 내용 및 검사 도구의 이해를 돕기 위해 연구에 참여하는 담임교사를 대상으로 사전 교육을 실시하였다. 본 연구의 효과성을 검증하기 위해 사용되는 설문과 검사는 사전·사후 검사로 이루어져 있으며, 유아들이 처음 접하는 소프트웨어 교육이기 때문에 용어와 내용이 익숙하지 않는 점을 고려해 3차시까지 수업을 진행 후 사전 검사를 실시하였다. 사후 검사는 10차시 수업이 끝난 후 진행하였으며 설문 중 결측치는 발견되지 않았다.

본 연구의 실시 기간은 2019년 8월이고, 연구대상자들은 소프트웨어 교육에 대해 경험이 없는 대상자이며, 연구 대상자의 일반적 특성은 만 5세(2013년 1월~12월생) 유아로 2개 반 총 41명 중 남아 19명(46.3%), 여아 22명(53.7%)이다.

2) 연구 도구

본 연구에 사용되는 ‘누리과정과 연계한 유아 소프트웨어 교육 프로그램1)’은 연구자가 개발한 교육 목적과 내용체계, 프로그램 예시를 바탕으로 유아교육 교수, 유아교육 전문가들이 개발하였으며 만 5세를 대상으로 하고 디지털 컬러, 알고리즘 라이프, 플레이 컴퓨팅으로 구분된 총 30개의 프로그램으로 구성되어 있다. 수업의 과정은 문제 발견/이해, 자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘, 자동화, 시뮬레이션으로 구성되어 있고, 소프트웨어 교육 내용 요소에는 정보윤리, 정보기기와 소프트웨어, 협업, 추상화, 알고리즘, 자료, 프로그래밍으로 구성되어 있다. 누리과정과 잘 어울리면서 실생활의 다양한 경험을 바탕으로 놀이와 체험을 통해 컴퓨팅 사고력을 향상할 수 있는 교육 프로그램으로 개발되었다. <표 IV-1>은 개발된 전체 프로그램의 활동에 대한 내용이다.

<표 IV-1> 소프트웨어 교육 프로그램

차시	구분	수업의 과정→								소프트웨어 교육 내용요소	
		문제 발견/이해	자료 수집	자료 분석	자료 표현	문제 분해	추상화	알고리즘	자동화		시뮬레이션
디지털 컬러	1. 나를 보호해요 ① - 나를 나타내는 것들이 있어요!	→	→	→							정보윤리 정보기기와 소프트웨어
	2. 나를 보호해요 ② - 그 다음에 어떤 일이 벌어질까?	→	→	→							정보윤리 정보기기와 소프트웨어
	3. 그림말(이모티콘)을 만들어요!	→	→	→			→			→	정보윤리 정보기기와 소프트웨어 협업
	4. 온라인 칭찬 릴레이	→	→		→					→	정보윤리 정보기기와 소프트웨어
	5. 구연동화를 만들어요!	→				→	→	→	→	→	정보기기와 소프트웨어 협업, 추상화
	6. 동생반을 위한 영상 만들기 ① - 동영상을 어떻게 만들까?	→	→	→		→		→			정보윤리 협업, 알고리즘
	7. 동생반을 위한 영상 만들기 ② - 스마트폰 올바른 사용 습관 영상을 촬영해요	→		→	→			→			정보기기와 소프트웨어 협업
	8. 동생반을 위한 영상 만들기 ③ - 온라인으로 동영상을 전해요	→	→	→	→	→		→			정보윤리 협업

1) 대한민국 교육부와 한국연구재단의 지원을 받아 수행중인 ‘누리과정을 연계한 유아 소프트웨어 교육 프로그램 개발’ 연구 과제를 통해 유아교사와 유아교육 전문가가 개발한 유아 소프트웨어 교육 프로그램

차시	구분	수업의 과정→								소프트웨어 교육 내용요소	
		문제 발견/ 이해	자료 수집	자료 분석	자료 표현	문제 분해	추상화	알고리즘	자동화		시뮬레이션
알고리즘 라이프	9. 동물 사진 전시회 ① - 내가 찾는 동물 사진은?	→	→	→				→			정보기기와 소프트웨어 자료
	10. 동물 사진 전시회 ② - 사진을 보기 좋게 전시하는 방법은?		→	→	→					→	협업, 자료
	11. 우리 동네 즐기기 ① - 우리 동네를 새로운 친구에게 소개하려면?	→	→	→	→						협업, 자료
	12. 우리 동네 즐기기 ② - 심심할 때 어디로 갈까?	→	→	→				→		→	알고리즘 프로그래밍
	13. 이슬이의 심부름을 어떻게 도 울까?	→				→		→		→	협업, 추상화
	14. 만약에 그래도 해결되지 않으 면?	→	→				→	→		→	추상화, 알고리즘 프로그래밍
	15. 문어가 행복한 표정이 되려 면?						→	→	→	→	추상화, 알고리즘 프로그래밍
	16. 생일 파티 준비하기 ① - 어떤 생일파티를 하고 싶니?	→	→	→		→	→	→			자료, 추상화, 알고리즘
	17. 생일 파티 준비하기 ② - 교실을 파티 장소로 꾸미려 면?			→		→	→	→			협업, 자료, 추상화
	18. 장난감을 잘 고르려면?	→				→	→	→			추상화, 알고리즘
플래이 컴퓨팅	19. 우리 반의 김밥 요리법 ① - 나만의 김밥을 만드는 방법 은?	→		→		→	→	→			추상화, 알고리즘
	20. 우리 반의 김밥 요리법 ② - 우리가 가장 좋아하는 김밥 은?	→		→			→	→			협업, 추상화 알고리즘
	21. 순환운동 만들기 ① - 한 바퀴 운동을 만드는 방법 은?	→	→	→	→	→					협업, 추상화 언플러그드/SW컴퓨팅
	22. 순환운동 만들기 ② - 제자리에서 할 수 있는 운동 은?	→	→	→	→	→	→				자료, 추상화 언플러그드/SW컴퓨팅
	23. 순환운동 만들기 ③ - 우리가 만든 한 바퀴 운동					→	→	→	→	→	알고리즘 언플러그드/SW컴퓨팅
	24. 순환운동 만들기 ④ - 한 바퀴 운동, 해보니 어땠나 요?								→	→	협업, 알고리즘 언플러그드/SW컴퓨팅

차시	구분	수업의 과정→									소프트웨어 교육 내용요소
		문제 발견/ 이해	자료 수집	자료 분석	자료 표현	문제 분해	추 상 화	알 고 리즘	자 동 화	시 물 레이 션	
플 래 이 컴 퓨 팅	25. 보드게임 만들기 ① - 우리가 좋아하는 보드게임은?	→	→	→	→						협업, 자료 언플러그드/SW컴퓨팅
	26. 보드게임 만들기 ② - '친구를 찾아라' 게임을 만드는 방법은?				→	→	→	→			추상화, 알고리즘 언플러그드/SW컴퓨팅
	27. 보드게임 만들기 ③ - 게임을 해보고 평가해요							→	→	→	협업 언플러그드/SW컴퓨팅
	28. 로봇 프로그래밍하기 ① - 모래시계가 불편해요	→	→	→	→						정보기기와 소프트웨어 자료, 피지컬 컴퓨팅
	29. 로봇 프로그래밍하기 ② - 로봇에게 무엇을 명령할까?				→	→	→	→	→		추상화, 피지컬 컴퓨팅
	30. 로봇 프로그래밍하기 ② - 양치질을 도와줘, 로봇!	→								→	프로그래밍 피지컬 컴퓨팅

(1) 교육 프로그램 구성

본 연구에서는 10차시 활동 과정에서 소프트웨어 교육의 궁극적인 목적을 달성할 수 있도록 소프트웨어 교육 내용요소와 컴퓨팅 사고력 요소가 균형 있게 분포하고 있는 13개의 프로그램을 선정하였다. 선택된 10차시의 소프트웨어 교육 프로그램과 수정된 활동명, 주요활동 내용을 살펴보면 <표 IV-2>와 같다.

<표 IV-2> 소프트웨어 교육 프로그램 10차시

차시	구분	활 동 명	수업의 과정→									소프트웨어 교육 내용요소
			문제 발견/ 이해	자료 수집	자료 분석	자료 표현	문제 분해	추 상 화	알 고 리즘	자 동 화	시 물 레이 션	
1차시	디 지 털 컬 쳐	1. 나를 보호해요 ① - 나를 나타내는 것들이 있어요!	→	→	→							정보윤리 정보기기와 소프트웨어
		주요활동 내용	- 개인정보가 무엇인지 알고 중요성에 대해 이야기 나누기 - 비밀번호가 필요한 상황 알고 직접 나만의 비밀번호 만들어 보기									
		2. 나를 보호해요 ② - 그 다음에 어떤 일이 벌 어질까?	→	→	→							정보윤리 정보기기와 소프트웨어
		주요활동 내용	- 개인정보가 유출되면 발생할 수 있는 일에 대해 이야기 나누기 - 개인정보를 보호할 수 있는 방법과 가정에서 실천하는 방법									
2차시		4. 온라인 칭찬 릴레이	→	→		→					→	정보윤리 정보기기와 소프트웨어

차시	구분	활동명	수업의 과정→								소프트웨어 교육 내용요소		
			문제 발견/ 이해	자료 수집	자료 분석	자료 표현	문제 분해	추상 화	알고 리즘	자동 화		시물 레이 션	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 우리 유치원 홈페이지와 블로그 살펴보기</li> <li>- 온라인에서 악성 댓글을 본 경험에 대해 이야기 나누기</li> <li>- 온라인 카페에서 친구 칭찬해 보기</li> <li>- 친구와 등 돌리고 앉아 서로 말로하는 댓글 경험해보기</li> </ul>										
3차시		13. 이슬이의 심부름을 어떻게 도울까?	→				→					협업, 추상화	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 이슬이의 문제 상황 이해하기</li> <li>- 심부름 중에 벌어질 수 있는 어려운 일들 골라 3단계로 해결해보기</li> <li>- 해결한 방법에 대해 친구들에게 소개하기</li> </ul>										
4차시	알고리즘 라이프	15. 문어와 꿀벌이야기						→	→	→	→	추상화, 알고리즘 프로그래밍	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 문어와 꿀벌의 해결하고자 하는 문제 인식하기</li> <li>- 문제 해결 절차를 간단한 그림이나 기호로 표현해보기</li> <li>- code.org를 통해 프로그래밍 도구 경험해보기</li> </ul>										
5차시		16. 어떤 생일파티를 하고 싶나?	→	→	→		→	→	→			자료, 추상화 알고리즘	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 생일 파티를 위해 필요한 자료 찾아보기</li> <li>- 생일 파티를 위한 순서 생각해서 적어보기</li> <li>- 필요한 자료와 필요하지 않는 자료 구분하기</li> </ul>										
6차시		19. 나만의 김밥을 만드는 방법은?	→		→		→	→	→			추상화, 알고리즘	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 김밥 요리 순서를 통해 단계적으로 생각하기</li> <li>- 요리 순서 각각의 상황에 생길 수 있는 문제 예측해보고 해결방법에 대해 이야기나누기</li> <li>- 요리 순서도 만들기</li> </ul>										
7차시	플래이 컴퓨팅	21. 순환운동 만들기 ① - 한 바퀴 운동을 만드는 방법은?	→	→	→	→	→					협업, 추상화 언플러그드/SW컴퓨팅	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 순환 운동을 위해 순서 정하기(순서도 그려보기)</li> <li>- 제자리에서 할 수 있는 다양한 동작 만들어보기</li> </ul>										
		22. 순환운동 만들기 ② - 제자리에서 할 수 있는 운동은?	→	→	→	→	→	→					자료, 추상화 언플러그드/SW컴퓨팅
		주요활동 내용	<ul style="list-style-type: none"> <li>- 순환카드 그리기</li> <li>- 순환카드를 패턴으로 나열해 운동 코스 만들기</li> </ul>										
		23. 순환운동 만들기 ③ - 우리가 만든 한 바퀴 운동						→	→	→	→	→	알고리즘 언플러그드/SW컴퓨팅
		주요활동 내용	<ul style="list-style-type: none"> <li>- 소그룹으로 나열된 순환카드의 운동 직접 해보기</li> <li>- 문제점이 있는지 확인하고 수정해보기</li> <li>- 다양한 방법의 순환카드 나열 후 직접 해보기</li> </ul>										
8차시		28. 로봇 프로그래밍하기① - 생활 속에서 함께하는 로봇들 알아보기	→	→	→	→						정보기기와 소프트웨어 자료, 피지컬 컴퓨팅	

차시	구분	활동명	수업의 과정→								소프트웨어 교육 내용요소	
			문제 발견/ 이해	자료 수집	자료 분석	자료 표현	문제 분해	추상 화	알고 리즘	자동 화		시물 레이 션
		주요활동 내용	<ul style="list-style-type: none"> <li>- 로봇을 본 경험과 어떻게 움직이는지 나의 생각 이야기 해보기</li> <li>- 로봇에게 명령을 내려 알람시계 만들어 보기</li> <li>- 소그룹 활동을 통해 신호등 만들어 보기</li> </ul>									
9차시		29. 로봇 프로그래밍하기 ② - 로봇에게 무엇을 명령할 까?				→	→	→	→	→	추상화, 피지컬 컴퓨팅	
		주요활동 내용	<ul style="list-style-type: none"> <li>- 로봇에게 어떤 기능을 명령할 수 있는지 이야기 해보기</li> <li>- 명령카드로 로봇이 반복하게 명령해보기</li> <li>- 소그룹 활동을 통해 로봇과 그대로 멈춰라! 놀이 해보기</li> </ul>									
10차시		30. 로봇 프로그래밍하기 ③ - 길을 찾아가는 걸 도와 줘, 로봇!	→								→	프로그래밍 피지컬 컴퓨팅
		주요활동 내용	<ul style="list-style-type: none"> <li>- 언플러그드 활동을 통해 친구에게 길을 찾아갈 수 있도록 명령 내려 보기</li> <li>- 로봇에게 명령을 내려 길을 찾아갈 수 있도록 해보기</li> </ul>									
<b>컴퓨팅 사고력 요소(포함/전체)</b>			8/22	5/16	5/18	4/11	5/13	5/13	6/18	3/6	5/12	

(2) 교육 프로그램 예시

선택된 10차시 중 이론 위주로 구성된 내용 부분을 소프트웨어 교육 관점에서 활동으로 변경하여 재구성하였으며 소프트웨어 전문가 3인의 타당도 검증 과정을 거쳤다. 2주간 매일 방과후 과정 오후 시간에 하루 1차시 30분~50분 동안 연구를 진행하였다. 시간은 프로그램의 목적과 유아들의 반응에 따라 융통성 있게 조절하였으며 교육은 소프트웨어 전문 교사인 연구자가 직접 진행되었다. 재구성된 교육 프로그램 예시를 살펴보면 [그림 IV-1], [그림 IV-2]와 같다.

주요개념 표기를 통해 소프트웨어 교육 내용요소 중 어디에 포함되는 제시해 주고, 생활주제는 누리과정 관련 생활주제와 어떻게 연결되는지 표시해 주었다. 활동목표, 누리과정 관련요소, 소프트웨어 교육과정 관련요소를 자세하게 표시해 교육활동을 통해 도달해야 하는 목표가 무엇인지 소프트웨어 교육과정의 어떤 요소에 해당하는지를 정확하게 확인하고 교육을 진행할 수 있도록 구성하였다. 활동자료에는 활동에 필요한 자료가 무엇인지를 확인해 미리 준비할 수 있도록 하였다. 활동방법에는 수업 과정들이 컴퓨팅 사고력 요소의 어디에 해당되는지를 분류해 표시하고 활동 과정안을 요약하여 제시하였다.

### 29. 로봇에게 무엇을 명령할까? (로봇 프로그래밍하기 ②)

생활주제	생활도구	활동형태	대·소집단활동-과학					
<b>활동목표</b>	<ul style="list-style-type: none"> <li>· 생활을 더 편리하게 하기 위해 로봇을 사용해보는 경험을 한다.</li> <li>· 여러 가지 명령카드 중 나에게 필요한 것을 선택하여 사용할 수 있다.</li> <li>· 로봇에게 원하는 일을 명령하기 위해 순서를 생각하며 코딩해본다.</li> </ul>							
<b>누리과정 관련요소</b>	<ul style="list-style-type: none"> <li>· 의사소통 &gt; 듣기 &gt; 바른 태도로 듣기</li> <li>· 사회관계 &gt; 다른 사람과 더불어 생활하기 &gt; 공동체에서 화목하게 지내기</li> <li>· 자연탐구 &gt; 과학적 탐구하기 &gt; 간단한 도구와 기계 활용하기</li> </ul>							
<b>소프트웨어 교육과정 관련요소</b>	<ul style="list-style-type: none"> <li>· 알고리즘라이프 &gt; 추상화 &gt; 해결하고자하는 문제를 인식하기</li> <li>· 플레이컴퓨팅 &gt; 피지컬 컴퓨팅 &gt; 로봇의 사용 방법 및 주의 사항 알기</li> <li>· 플레이컴퓨팅 &gt; 피지컬 컴퓨팅 &gt; 프로그램에 따른 로봇 동작 확인하기</li> </ul>							
<b>활동자료</b>	코딩로봇, 명령카드							
활동방법								
<b>문제탐색예</b>	<b>자료수집</b>	<b>자료분석</b>	<b>자료표현</b>	<b>문제분해</b>	<b>추상화</b>	<b>알고리즘</b>	<b>자동화</b>	<b>시뮬레이션</b>
<p><b>1. 로봇을 이용하기 위해 무엇을 명령할지 생각해본다.</b> <span style="background-color: #cccccc;">자료표현</span></p> <ul style="list-style-type: none"> <li>- 로봇에게 어떤 것을 명령하고 싶니?</li> <li>예) 그대로 멈춰라 놀이</li> </ul> <p><b>2. 로봇에게 어떤 기능을 명령할 수 있는지 알아본다.</b> <span style="background-color: #cccccc;">문제분해</span></p> <ul style="list-style-type: none"> <li>- 어떤 것들을 명령해야 로봇이 멈추지 않고 계속해서 움직일 수 있을까?</li> </ul> <p><b>3. 우리가 원하는 기능 중 로봇에게 명령할 수 있는 기능을 찾아본다.</b> <span style="background-color: #cccccc;">문제분해</span></p> <ul style="list-style-type: none"> <li>- 이 중에서 우리에게 필요한 명령카드는 무엇일까?</li> <li>- 우리가 하고 명령하고 싶지만, 명령카드가 없어서 못하는 것도 있을까?</li> </ul> <p><b>4. 로봇에게 명령카드로 코딩해본다.</b> <span style="background-color: #cccccc;">추상화</span> <span style="background-color: #cccccc;">알고리즘</span></p> <ul style="list-style-type: none"> <li>- 로봇이 앞뒤로 움직이게 하려면 어떤 명령카드를 사용해야 할까?</li> <li>- 로봇에게 손을 가져가면 멈추게 하려면 어떤 명령카드를 사용해야 할까?</li> <li>- 이렇게 명령하고 싶은 카드를 순서대로 놓아 보자.</li> </ul>								

[그림 IV-1] 유아 소프트웨어 교육 프로그램 예시-1

참고사항에는 교육을 위한 사전 교육이 필요한 경우와 수업 시 주의해야 할 사항에 대한 내용이 포함되어 있다. 가정과의 연계에는 교육기관 내에서의 교육으로 그치지 않고 가정에서 부모와 함께 할 수 있는 내용, 교육 후 가정에서 예

상 질문과 답변, 주의사항들에 대한 내용들이 포함되어 있다. 선생님을 위한 도움말에는 유아 교사가 교육을 진행할 경우 전문지식이 부족해 알려주지 못할 수 있는 내용에 대해 요약 내용을 추가하여 이해를 도왔다. 이를 통해 교육 속에 컴퓨팅 원리를 자연스럽게 녹여내 간접적이지만 최대한의 교육적 효과를 기대할 수 있도록 하였다.

<p><b>5. 명령한 기능을 반복되게 하려면 어떻게 해야 할지 알아보고 명령한다.</b> <span style="float: right;">[자동화]</span></p> <ul style="list-style-type: none"> <li>- 로봇이 앞으로 움직이는 걸 계속 반복하게 하려면 어떤 명령카드를 사용해야 할까?</li> <li>- 계속 반복하는 걸 멈출 필요가 있을 때는 어떤 명령카드를 사용해야 할까?</li> </ul> <p><b>[소그룹 활동]</b></p> <p><b>6. 그대로 멈춰라! 놀이를 떠올려보며 코딩이 잘 되었는지 확인해본다.</b></p> <p><span style="border: 1px solid black; padding: 2px;">알고리즘</span> <span style="border: 1px solid black; padding: 2px;">자동화</span></p> <ul style="list-style-type: none"> <li>- 머릿속으로 그대로 멈춰라! 놀이를 상상하며 카드를 순차적으로 놓고 로봇에게 입력했을 때 어떻게 움직일지 생각해보자.</li> <li>- 로봇에게 순차적으로 카드를 입력하고 결과를 확인해보자.</li> <li>- 직접 사용해보고, 어떤 점이 잘됐는지, 힘들었던 점은 무엇이었는지 찾아보기로 하자.</li> <li>- 다른 놀이를 로봇과 함께 한다면 어떤 걸 하고 싶은지 이야기 해 보자.</li> </ul>	
<b>참고사항</b>	<ul style="list-style-type: none"> <li>- 본 활동을 하기 전에 미리 자유선택활동 시간에 로봇에게 명령카드를 입력해보는 경험을 충분히 할 수 있도록 한다.</li> <li>- 먼저 각 명령 카드의 기본 기능을 익히고, 반복하기 기능 등을 사용해보도록 순차적으로 접근한다.</li> <li>- 유아들과 함께 정한 코딩 순서는 사진으로 찍거나, 기록해두어 다음 시간에 이어서 수업할 때 사용한다.</li> </ul>
<b>가정과의 연계</b>	<p style="text-align: center; border: 1px solid gray; padding: 5px;">유아기, 로봇과의 첫 만남은 어떻게?</p> <ul style="list-style-type: none"> <li>* 유아부터 초·중등학교 학생들을 대상으로 한 다양한 코딩로봇들이 개발되고 있어요. 코딩로봇의 사용방법을 가르쳐주는 곳도 많지요. 그런데 로봇 자체가 학습의 목적이 될 수는 없어요. 즉, 우리는 로봇을 익히기 위해서 공부하고 사용하는 것이 아니라, 우리 생활을 더 편리하게 하는 방법을 생각하다 보니 로봇을 이용하고 싶어지고, 그래서 로봇을 다루는 법을 배우게 되는 것입니다.</li> <li>* 가정에서도 아이들이 로봇을 만날 기회(예: 로봇청소기로 청소하기)가 있다면, 로봇에게 무엇을 명령하면 우리 생활이 더 즐겁고 편해질까, 어떤 기능이 더해진다면 좋을까 등의 궁극적 목적부터 인식하도록 지도하는 것이 중요합니다.</li> </ul>
<b>&lt;선생님을 위한 도움말&gt;</b>	
<p><b>♣ 코드(Code)</b></p> <p>코드는 컴퓨터가 알아들을 수 있는 언어로, 컴퓨터가 해야 할 일을 알려주는 명령들을 모아놓은 것을 말합니다. 광범위하게 보면 특정한 형태의 정보를 표현하는 규칙으로, 우리가 사용하는 언어도 코드의 일종이라고 볼 수 있지요. 또 바코드, QR코드 등도 우리에게 친숙한 코드의 종류입니다.</p> <p>그리고 컴퓨터는 0과 1 두 가지 숫자를 조합한 이진 코드를 사용해요. 자료를 처리하기 위해 일정한 규칙에 따라 코드를 만드는 것이 코딩이며, 컴퓨터가 복잡한 일을 빠르게 처리할 수 있게 코딩하려면 먼저 알고리즘을 구체적으로 만드는 일이 중요해요.</p>	

[그림 IV-2] 유아 소프트웨어 교육 프로그램 예시-2

### (3) 타당도 검증

수정된 교육 프로그램을 연구도구로 사용하기 전 타당성 검증을 위하여 프로그램 내용에 대해 전문가 인터뷰 과정을 거쳤다. 수정된 교육 프로그램은 대학 교수로 구성된 소프트웨어 전문가 3인이 타당성 검토를 시행하였다. 전문가들은 유아들에게 교육하기 적합하게 수정이 되었는지, 자연스럽게 소프트웨어 교육이 이루어지도록 수정되었는지, 문제해결능력, 컴퓨팅 사고력을 향상시킬 수 있는 교육 프로그램으로 구성되어 있는지, 피지컬 도구를 활용하여 소프트웨어를 체험하는 놀이형태의 교육으로 구성되어 있는지, 추가적으로 포함해야 하거나 수정이 필요한 내용이 있는지 등을 중점으로 검토하였다.

이 과정을 통해 얻어진 교육 프로그램에 대한 전문가들의 주요 의견들을 살펴보면 다음과 같다.

첫째, 수정된 프로그램은 디지털 컬러와 함께 소프트웨어 교육을 다양한 경험을 통해 자연스럽게 이루어지도록 구성되었다. 대부분의 학습 목표가 유아에게 적합한 경험하거나, 사용하거나, 알아보거나, 생각해 볼 수 있도록 적절하게 제시되고 있다.

둘째, 활동 방법이 주로 질문과 대답으로 이루어진 교수법으로 유아들이 스스로 생각하고 문제를 직접 해결해 볼 수 있도록 유도하고 스스로 논리적인 사고를 할 수 있도록 구성되어 있다. 교수자는 옆에서 조력자 역할을 할 수 있도록 프로그램이 구성되어 있어 유아들의 문제해결 능력, 논리적 사고를 충분히 향상시킬 수 있는 교육 프로그램이다.

셋째, 기존 프로그램에 비해 소그룹 활동을 포함한 세부적인 단계가 제시되어 유아에게 교육하기에 적합하다. 소그룹 활동을 통해 명령카드를 익히고 알버트에게 입력하는 과정이 잘 나타나있는 놀이형태 교육이다.

넷째, 명령카드를 이해한 것과 로봇에게 입력한 결과의 차이를 유아들이 스스로 이해하고 판단할 수 있도록 자기 주도적 형태의 학습이 이루어지도록 해야 한다. 명령카드의 순서에 따라 다양한 해결책이 나올 수 있기 때문에 유아들의 의도와는 상관없이 창의적인 해결책이 나오더라도 격려해 주고, 스스로 생각할 수 있도록 칭찬 요소를 보완해야 한다.

다섯째, 교사가 컴퓨팅 사고에 대해서 올바르게 이해하고 가르칠 수 있도록 각

활동은 일관성을 가지고 설명될 수 있도록 주의해야 한다. ‘문제발견/이해’와 ‘추상화’의 활동이 구분되어야 하며 ‘시뮬레이션’은 상상하는 것이 아니라 자동화를 통해 시뮬레이션 하는 영역이다.

전문가 의견을 통한 시사점을 도출해보면 학습목표는 유아 수준에 적절하게 제시되어 있으며 스스로 생각하고 사고하여 문제를 직접 해결해 볼 수 있도록 구성되어 있다. 기존 프로그램에 비해 소그룹 활동을 포함하여 세부적인 단계가 잘 제시되어 교육하기에 적합하다. 컴퓨팅 사고력 요소에 해당하는 내용들을 하나로 묶으려고 하지 말고 여러 가지로 나열하는 방법이 필요하다. 협의된 내용들을 수렴하여 교육 프로그램의 내용 및 활동방법을 수정·보완하고 이를 바탕으로 교육 현장에 적용 가능한 프로그램이 최종 완성되었다.

### 3) 검사 도구

평가에 대한 개념은 교육평가의 강조점이 어디에 있느냐에 따라 다르게 분류된다. 첫째, 교육목표 달성도 확인에 초점을 두었을 때, 평가란 교육 과정과 수업 활동을 통해 교육목표가 얼마나 달성되었는가를 판단을 내리는 것으로 정의할 수 있다(Tyler, 1949). 둘째, 의사결정을 내리는 것에 초점을 두었을 때, 평가란 정보를 수집하고 사용을 통해 수업 개선에 관한 의사결정, 학생 개인에 관한 의사결정, 행정 체제에 관한 의사결정을 하는 과정으로 정의할 수 있다(Cronbach, 1984). 셋째, 통합적·총론적으로 교육의 과정이나 결과에 대해 어떤 결정을 내릴 목적으로 교육의 과정과 결과에 대한 가치와 장점을 체계적으로 조사·활용하는 것이다(Nevo, 1986). 따라서 평가란 교육목표의 달성 정도를 판단하고 교육활동의 성과를 가늠하여 결과에 대한 가치를 판단하는 일련의 과정이라고 할 수 있다.

#### (1) 평가 목적

평가의 목적은 학습자의 교육목표 도달도를 확인하여 학습과 수업의 질을 개선하기 위함이다. 평가의 내용은 교육내용과 기능 즉, 성취기준이며 학습의 결과 뿐 아니라 과정도 평가해야 한다(고영희 외, 2018). 그 중 유아의 평가는 유아의 발달정도, 특정 유아교육 프로그램과 유아와 상호작용의 효과, 유아 개개인의 능력 및 특성과 잠재력에서 개인차 등을 평가하여 교육에 반영시킴으로써 개인 발

달의 최대화를 도모하기 위한 것이다(최일선 & 조운주, 2010) 즉 교육 평가는 교육 목표가 올바르게 설정되었는지, 목표를 실현하기 위한 교육을 위한 계획과 과정은 적절한지, 그리고 궁극적으로 교육의 목표가 제대로 성취되었는지를 확인하고 판단하는 일련의 과정이라 할 수 있다(장성모, 1998).

유아 및 아동 발달연구에서 평가의 목적과 방법에 따라 다양하게 적용하고 있지만 일반적으로 사용하는 자료수집 방법에는 관찰법, 질문지법, 검사법, 면접법 등이 있다.

관찰법(observation)은 일상적인 상황에서 자연스럽게 나타나는 연구대상의 행동을 관찰하고 기록하여 객관적인 자료를 수집하는 방법이다. 일반적으로는 유아의 발달을 알아보기 위해 관찰을 하지만 교사들의 교수행동, 교사와 유아의 상호작용, 교육환경 등 유아와 학습의 모든 대상과 장면, 상황 모든 것이 관찰의 대상이 될 수 있다(황해익, 2009). 표현이 미숙한 연구대상에게 유용하게 적용할 수 있는 평가 방법으로 역사가 상당히 오래되었다. 다른 자료수집 방법들에 비해 관찰대상의 협력을 비교적 덜 필요로 하는 반면 관찰자는 고도의 전문적인 능력을 요구하는 방법이다.

질문지법(questionnaire)은 연구하고자 하는 내용에 관련된 질문을 먼저 작성하고 이를 연구 대상에게 답하도록 하여 그 내용들을 분석하는 방법이다. 질문지법에는 선다형, 평정척도형, 순위형, 분류형, 자유반응형 다섯 가지 형식이 있다. 다수를 대상으로 많은 자료를 손쉽게 짧은 시간 안에 수집할 수 있는 장점이 있으며 발달에 관련된 연구에 가장 많이 사용되는 방법이다. 그러나 문자를 읽기 어려운 유아에게 사용하기 어렵고 가정 등으로 전달되었을 경우에는 회수율이 낮다는 한계점이 있다.

검사법은 표준화된 검사를 이용해 연구 대상의 특성에 관한 자료를 수집하는 방법이다. 20세기 들어서 적성, 지능, 성격, 흥미, 태도, 능력 등을 측정하기 위한 다양한 종류의 심리검사가 개발되어 사용되고 있다. 검사법은 개인 차이의 예측과 측정, 진단이 가능해 결과를 통해 발달 특성을 이해할 수 있을 뿐만 아니라, 또래와의 상대적 비교도 가능하여 객관적이고 구체적인 정보를 얻을 수 있는 장점이 있다. 그러나 나이가 어린 연구대상에게 적용하기 위해서는 신뢰도와 타당도가 충분히 확보된 검사를 신중하게 선택하여 실시하여야 할 필요성이 있다.

면접법(interview)은 연구대상에게 직접 질문을 하고 응답을 기록해 자료를 수집하는 방법이다. 면접 시 행동까지 관찰이 가능하며 응답에 따라 추가 질문을 할 수 있어 보다 자세한 자료를 수집할 수 있다는 장점이 있어 영유아 및 아동 발달연구에 자주 사용되는 자료 수집 방법이다. 그러나 면접자의 주관적 해석이나 면접자의 조건에 따라 결과가 달라질 수도 있고 일대일로 진행하기 때문에 시간과 노력이 많이 든다는 단점이 있다.

## (2) 평가 내용과 방법

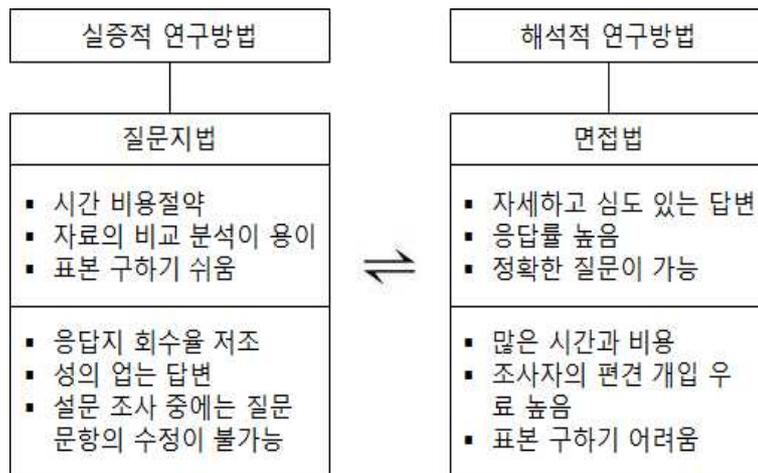
이러한 내용을 종합하여 유아 소프트웨어 교육을 위한 평가 방법에는 검사법과 질문지법+면접법을 선정하였다.

유아 평가는 유아의 학습 성취 여부에만 초점을 두기보다는 유아교육 활동의 전체적인 과정과 결과에 대한 가치판단이 이루어져야 함을 의미한다. 평가 방법이 유아의 발달과 경험에 적합하고, 평가의 목적과 본질이 유아에게 유익함을 주어야 한다(최일선 & 조운주, 2010). 반드시 프로그래밍이 포함되어야 하는 것은 아니며, 프로그래밍을 하지 않더라도 컴퓨터과학의 개념과 원리를 활용하여 문제를 효과적으로 해결하였다면 컴퓨팅 사고력에 포함될 수 있다(최정원 & 이영준, 2014). 양재명, 김자미, 이원규, 윤일규, 서정희(2017)의 초·중등 소프트웨어 교육 역량 진단도구 개발 연구에서는 실생활에서 경험할 수 있는 내용으로 구성하고, 문제 상황에 따른 하위 문항을 구성하여 확장할 필요성이 있다고 보았다. 또한 실생활의 소재로 시나리오 구성, 전문적 개념을 포함하지 않고 사고력만으로 문제를 해결할 수 있도록 구성, 문제해결과정의 사고력을 측정, 사고의 흐름을 고려한 문항, 학령기에 적합한 어휘 수준 등이 고려되어야 한다.

유아 소프트웨어 교육의 목적은 일상생활 속의 문제 해결을 중심으로 이루어진 활동을 통한 컴퓨팅 사고력 함양이다. 소프트웨어 교육에서의 평가는 단순히 컴퓨팅 사고력을 측정하는 것뿐만 아니라 평가를 통해 컴퓨팅 사고력을 향상시킬 수 있다는 점에서 매우 중요하다. 이런 점에서 소프트웨어 교육에서 어떻게 컴퓨팅 사고력을 평가할 것인가를 고려한 다양한 연구가 이루어져야 할 것이다(최정원 & 이영준, 2014). 이러한 근거를 바탕으로는 유아 소프트웨어 교육 평가는 소프트웨어 교육 효과성 검사와 컴퓨팅 사고력 검사로 두 가지 형태로 이루어져야 한다고 제안한다.

자세한 평가 내용 및 방법을 살펴보면 다음과 같다.

첫째, 소프트웨어 교육 효과성에 대한 검사이다. 이를 통해 소프트웨어 교육의 목표가 성취되었는지의 여부를 밝혀 전반적인 교육활동을 개선하고 방향을 재설정 할 수 있다. 특히 교육 활동이 설정된 목표에 어느 정도 도달하였는지, 만약 제대로 달성되지 않았다면 그 요인은 무엇이며 어떻게 수정 보완이 필요한지에 대한 근거를 평가에서 찾을 수 있다(이경희 외, 2019). 이러한 평가는 질문지법+면접법이 가장 적합하다. 일대일로 진행하기 때문에 시간과 노력이 많이 들지만 유아들의 생각과 행동까지 관찰이 가능하다. 유아 소프트웨어 교육 효과성 검사는 교육이 유아들에게 미치는 영향과 효과를 알아보기 위해 박주연, 김종혜, 김석희, 이현숙, 김수환(2017)의 ‘초·중등 SW교육의 평가요소 개발’ 연구에서 제시한 문항 예시를 기초로 만 5세 유아의 수준과 연구의 목적에 맞게 수정·보완하여 질문지법+면접법으로 재구성하여 개발하였다. 질문지법으로 개발되었으나 글을 읽지 못하는 유아를 배려해 읽어주는 면접법도 함께 병행하였다. [그림 IV-3]은 질문지법과 면접법의 상호보완적인 관계에 대한 내용이다. 실증적 연구방법에 사용되는 질문지법과 해석적 연구방법에 사용되는 면접법을 같이 사용함으로써 각각의 검사 방법의 장단점을 보완할 수 있는 형태로 개발되었다.



[그림 IV-3] 질문지법+면접법

평가에는 교육의 전반적인 내용에 대한 가치, 태도, 컴퓨팅 사고력 효능감, 흥미 등이 포함된 내용이다. 가치 영역은 SW의 중요성, SW교육의 유용성, SW교

육 중요성, SW의 편리성, SW수업 필요성, SW필요성, SW수업 선호도, SW수업 내용에 대한 만족도, SW수업에 대한 기대감, SW관련 직업에 대한 인식, SW관련 전공에 대한 인식, SW교육에 대한 의지의 평가요소로 구성되어 있다. 태도 영역은 프로그래밍 몰입, 수업을 통한 의사소통, 친구와 상호작용, 교사와 상호작용, 수업에서의 협동심, 타인과의 협력을 통한 창의성, 공동체 의식, 프로그래밍 윤리 책임의 평가요소로 구성되어 있다. 컴퓨팅 사고력 효능감 영역은 과제해결에 대한 노력, 자신의 생각을 표현하는 능력, 프로그래밍 실행에 대한 탐구심, 적극적인 수업참여, 프로그래밍에의 자신감, 정보처리 사고과정의 자신감, SW수업을 통한 성취감의 평가요소로 구성되어 있다. 흥미 영역은 수업에 대한 흥미, 학습활동에 대한 흥미, 수업에 대한 지속의지, SW에 대한 관심의 평가요소로 구성되어 있다.

개발된 검사지는 유아교육 전문가 1인의 타당도 검증을 통해 유아가 알기 쉬운 단어로 수정·보완하여 완성하였다. 검사지는 3개의 영역과 13개의 세부영역으로 구성된다. 총 13개의 문항으로 구성되어 있으며 설문 문항이 유아가 생각하기에 정말 그렇다는 5점, 조금 그렇다는 4점, 보통이라는 3점, 그렇지 않다는 2점, 전혀 그렇지 않다는 1점으로 교육 활동 전과 활동 후 질문지법+면접법을 통해 실시하였다.

둘째, 컴퓨팅 사고력에 대한 검사이다. 2014년 영국에서 컴퓨팅 교육과정을 도입한 이후 본격적으로 컴퓨팅 사고력에 대한 평가가 논의되기 시작했다. 컴퓨팅 사고력 평가는 특정 검사 도구 사용, 설문조사, 관찰방법 3가지 유형으로 실시되고 있다(김민정 외, 2017). Brennan & Resnick(2012)는 컴퓨팅 사고력 평가에 대한 프레임워크에서 Scrape 도구를 사용한 자동화 평가 도구, 인터뷰를 통한 평가, 평가 틀을 이용한 방법 3개 유형의 평가 방법을 제시하였다. Seiter & Foreman(2013)은 스크래치 산출물을 통한 컴퓨팅 사고력 평가를 Evidence Variables, Design Pattern Variable, Computational Thinking Concepts 3가지 측면으로 평가하는 방법을 제안하였다. 소프트웨어 교육의 궁극적인 목적 달성 확인을 위해 컴퓨팅 사고력에 대한 평가는 반드시 이루어 져야 한다. 그러나 사고력 자체가 가지는 추상성 때문에 이를 측정할 표준화된 문항을 개발하는 것이 쉽지 않다(김민자외, 2016). 또한, 소프트웨어 교육의 목적과 내용, 교육방법에 따

라 평가문항 및 방법이 개발되어 평가가 이루어질 필요성이 있다.

컴퓨팅 사고력 검사는 소프트웨어 교육을 통해 컴퓨팅 사고력 향상 효과를 알아보기 위해 ‘Bebras Challenge - Castors : Years 4 & 5’(Bebras, 2018) 문제를 기초로 하여 한국 유아의 특성과 글자에 익숙하지 않는 유아들을 배려하여 그림 위주의 검사지로 재구성하여 개발하였다. 개발된 검사지는 유아교육 전문가 1인의 타당도 검증을 통해 유아의 특성 및 유아가 알기 쉬운 단어로 수정·보완하여 완성하였다. 각 문항은 컴퓨팅 사고력 요소인 문제 분석, 문제 분해, 추상화, 알고리즘, 자동화, 시뮬레이션과 컴퓨터과학 영역을 모두 포함하여 균형 있게 구성 되도록 하여 검사의 신뢰성을 확보할 수 있도록 하였다.

<표 IV-3> 컴퓨팅 사고력 검사도구 구성

	CT 요소	컴퓨터 과학 영역
문항 1	자동화, 알고리즘	알고리즘 및 프로그래밍
문항 2	알고리즘, 시뮬레이션	알고리즘 및 프로그래밍
문항 3	문제분석, 문제분해	알고리즘 및 프로그래밍
문항 4	추상화, 자동화, 시뮬레이션	컴퓨팅 시스템 구성 및 동작원리
문항 5	문제분석	자료분석 및 자료표현
문항 6	알고리즘, 자동화, 시뮬레이션	알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리
문항 7	문제분석, 문제분해	자료분석 및 자료표현
문항 8	알고리즘	알고리즘 및 프로그래밍
문항 9	알고리즘, 시뮬레이션	알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리
문항10	추상화	자료분석 및 자료표현

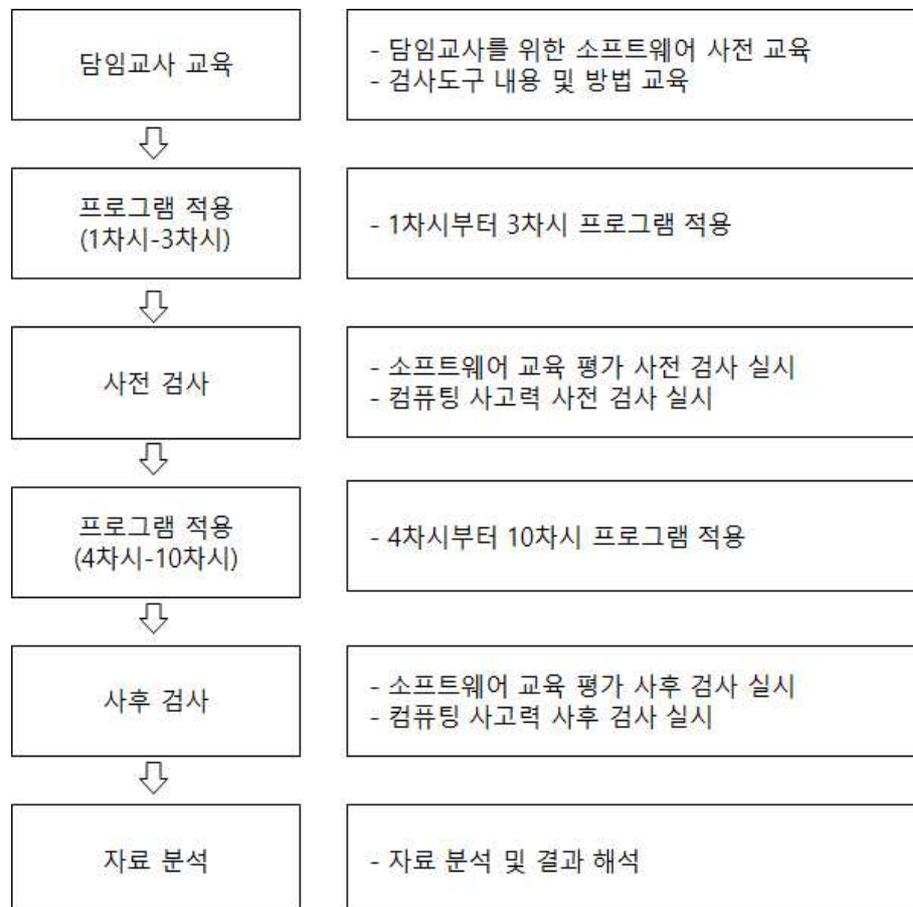
<표 IV-3>의 컴퓨팅 사고력 검사도구 구성을 자세히 살펴보면 검사지는 10개의 문항으로 구성되어 있으며 한 문항에 한 요인이 아니라 복합적으로 구성되어 있다. 일반적으로 4지 선다형으로 이루어져 있고, 몇 개의 문항은 그림을 그리거나 숫자를 적는 문항으로 구성되어 있으며 교육 활동 전과 활동 후 검사법을 통

해 실시하였다. 이는 점수나 등급을 매겨 우수한 유아를 구분 선발하기 위한 목적의 문제가 아니라 개개인의 컴퓨팅 사고력 측정을 강조하였다.

다만, 검사 내용에 소프트웨어에 대한 언급이 있어 활동 전 설문을 진행할 경우 유아들이 이해하지 못하는 것을 대비해 활동 전 검사의 경우 두 가지 검사 모두 3차시의 수업을 진행 한 후 실시하였다.

#### 4) 연구 절차

본 연구의 진행일정 및 절차는 [그림 IV-4]와 같다. 담임교사 교육을 시작으로 1~3차시 교육을 진행한 후 사전 검사를 실시하였다. 그 후 4~10차시 프로그램을 적용 후 사후 검사를 실시하고 자료 분석을 통해 결과를 해석하였다.



[그림 IV-4] 연구 진행 및 절차

(1) 교사교육

본 연구를 위해 교사교육을 실시하였다. 연구를 함께 진행할 교사는 각 반의 담임교사로 유치원 1급 정교사 자격증을 소지한 유아교육 전공의 현장 경력 10년차 이내 교사이다. 소그룹 활동에 투입을 대비해 방과후 전담교사도 함께 교육을 진행하였다. 먼저 연구의 목적 및 필요성을 설명하고, 교육의 목적과 본 연구에서 사용되는 용어 및 교육내용, 교구에 대한 설명과 소프트웨어 효과성 검사, 컴퓨팅 사고력 검사 진행 방법에 대하여 교육을 하였다. 교사교육은 프로그램 전반에 대한 내용은 인쇄물을 제시하면서 강의, 설명, 시연, 토론, 체험하는 방법으로 이루어졌다.

<표 IV-4> 사전 교사교육

구분	내용	자료
연구의 목적 및 필요성	<ul style="list-style-type: none"> <li>만 5세부터 시작하는 유아 소프트웨어 교육의 국가 교육과정 도입 시도</li> <li>유아 소프트웨어 교육을 통한 컴퓨팅 사고력 향상 확인</li> </ul>	인쇄물
교육 목적	<ol style="list-style-type: none"> <li>컴퓨팅 사고력(computational thinking)</li> <li>협력적 해결역량(Collaboration)</li> <li>정보문화 소양(Information Cultural Literacy)을 기르는 것</li> </ol>	인쇄물
활동 내용	<ul style="list-style-type: none"> <li>용어설명                             <ul style="list-style-type: none"> <li>소프트웨어 교육, 컴퓨팅 사고력, 협력적 해결역량</li> </ul> </li> <li>교육 진행방법 및 내용 설명                             <ul style="list-style-type: none"> <li>1차시~10차시 교육내용 설명</li> </ul> </li> <li>교구 조작방법 실습                             <ul style="list-style-type: none"> <li>code.org에 대한 교육</li> <li>로봇에 대한 교육</li> </ul> </li> </ul>	인쇄물 강의 시연 토론 체험
교수·학습 방법	<ul style="list-style-type: none"> <li>교육 진행 안내</li> <li>교육 진행 시 보조교사의 역할</li> </ul>	설명
사전 검사	<ul style="list-style-type: none"> <li>사전 검사의 목적 및 검사 방법 설명</li> <li>문항 설명과 평정 방법 설명</li> </ul>	인쇄물 토론
사후 검사	<ul style="list-style-type: none"> <li>사후 검사의 목적 및 검사 방법 설명</li> <li>문항 설명과 평정 방법 설명</li> </ul>	인쇄물 토론

교육 내용으로는 연구의 목적 및 필요성, 교육의 목적, 활동 내용, 교사의 역할, 사전 검사 및 사후 검사 방법과 이에 따른 유의사항 등을 교육하였다. 교사교육의 구체적인 내용은 <표 IV-4>와 같다.

담임교사의 역할은 소프트웨어 전문 교사가 알지 못하는 유아의 반응을 확인하여 상호작용할 수 있도록 도와주는 것이다. 소프트웨어 교육 효과성 검사와 컴퓨팅 사고력 검사를 함께 진행하며 유아들이 이해하기 쉽도록 단어나 내용에 대해 부연 설명해 주는 것이다.

### (2) 1차시~3차시 프로그램 적용

교사와 친밀한 관계를 형성하는 것이 교육에 있어 중요하기 때문에 프로그램 적용 전 오리엔테이션을 통해 유아와 친숙해지는 시간을 가졌다.

유아 소프트웨어 교육 활동 적용 시간은 1차시당 30~50분으로 프로그램의 특성과 유아의 반응에 따라 융통성 있게 조절되었으며 2주간 총 10회에 걸쳐 교육을 진행하였다. 내용에 따라 대집단 활동, 소집단 활동, 개별 활동으로 진행하였고 평균적으로 30분 정도 소요되었으며 소집단 활동은 추가적으로 10분에서 20분 정도 더 소요되었다. 처음 소프트웨어 교육을 접하는 유아들을 위해 검사 전 3회 교육을 진행하였으며 나를 나타내는 것들이 있어요!, 온라인 칭찬 릴레이, 이슬이의 심부름을 어떻게 도울까? 로 구성된 3개의 활동을 실시하였다. 교육 활동은 유아들의 관심과 흥미에 따라 진행하는 중간에라도 수정, 보완되어 진행하였다. 유아들은 영상, 동화책의 내용을 통해 정보윤리에 대해 배웠고 유아들이 흔히 보는 영상에 나오는 사용하지 않아야 하는 말들과 그 이유에 대해 활동을 진행했다. 이러한 활동에서 유아는 수업에 필요한 본인의 개인정보와 관련해 잘 알지 못해 원내에서 교육으로 그치지 않고 학부모와 함께하는 교육이나 가정과 연계할 수 있는 교육의 필요성을 시사하고 있었다.

### (3) 사전 검사

사전 검사는 3회 교육 후 실시하였다. 소프트웨어 교육 평가는 총 13개의 문항으로 구성된 질문지법+면접법으로 Likert 5점으로 평정한다. 검사 장소는 독립되고 안정된 조용한 장소에서 유아 한명씩 검사를 진행하였으며 글을 잘 읽지 못하는 유아 들을 배려해 교사가 질문을 읽어주고 유아의 대답에 따라 5점 척도로 구성된 답변에 체크하는 방법으로 진행하였다. 시간제한 없이 문항 중 단어의 뜻

또는 상황을 이해하지 못 할 경우 설명을 해주며 진행을 해 나가는 방식으로 한 명 당 5분 정도 소요되었다. 중간 중간 힌트와 독려를 통해 검사를 마칠 수 있도록 하였다. 컴퓨팅 사고력 검사는 이미지 위주로 구성된 총 10개의 문항으로 구성된 검사법으로 문항별 척도에 따라 평정한다. 검사 장소는 교실에서 반별로 진행하였으며 글을 잘 읽지 못하는 아이들을 배려해 질문을 교사가 먼저 읽어주고 설명 후 답을 작성하도록 하였다. 모든 검사는 신뢰도를 높이기 위해 소프트웨어 전문 교사와 담임교사가 함께 실시하면서 의견을 교환하였다. 컴퓨팅 사고력 검사 도구에 관한 평정 척도는 부록에 제시하였다.

#### (4) 4차시~10차시 프로그램 적용

4차시에서 10차시 프로그램은 사전 검사 이후에 진행이 되었으며 스마트 패드를 사용하는 문어와 꿀벌 이야기, 어떤 생일파티를 하고 싶니?, 나만의 김밥을 만드는 방법은?, 우리가 만든 한 바퀴 운동, 피지컬 컴퓨팅 교구를 사용하는 생활 속에서 함께 하는 로봇들 알아보기, 로봇에게 무엇을 명령할까?, 길을 찾아가는 걸 도와줘, 로봇! 으로 구성된 7개의 활동을 실시하였다. 스마트 패드를 사용해 코드닷오알지(<http://code.org/>) 사이트 내 교육은 소집단 활동으로 진행하였다. 알고리즘 라이프와 플레이 컴퓨팅이 포함되어 있는 차시에서는 최대한 소그룹 활동을 통해 협동학습을 많이 할 수 있도록 프로그램을 변경하였다. 협동학습은 유아의 문제해결 능력에도 영향을 미친다고 할 수 있다(Kamii, 1978; 1993). 이러한 근거는 Piaget의 이론에서 알 수 있듯이 Piaget는 문제에 관하여 서로 다른 견해를 가진 유아와의 갈등 경험을 탈 중심적 사고발달에 결정적인 요인으로 보고 있다(이수련, 2009). 또한, Vygotsky(2009)의 이론에 의하면 유아는 교사나 부모의 지도 보다 우수한 또래와의 ‘협동 작업’으로 현실적 발달 수준에서 잠재적 발달 수준으로까지 높아진다고 하였다. 이 과정에서 유아들에게 하나의 단계만 설명해 주었는데도 불구하고 소집단끼리 서로 협동하고 공유하며 시행착오를 거쳐 다음 단계로 진행을 해 냈다. 이를 통해 만 5세 연령이 협동학습, 협업, 공유가 가능하고 소프트웨어 교육에 이러한 방법이 효과가 있다는 것을 시사하고 있다.

#### (5) 사후 검사

사후 검사는 10회의 수업이 끝난 후 사전 검사와 동일한 방법으로 소프트웨어

교육 효과성 사후 검사, 컴퓨팅 사고력 사후 검사를 실시하였다. 교육 중간에 포기하는 유아가 없어 동일 대상, 동일 인원 41명의 검사가 이루어졌다. 사전 검사와 동일한 내용으로 이루어져 있기는 하나 사후 검사 역시 시간제한 없이 유아의 특성을 파악해 중간 중간 힌트와 독려를 통해 검사를 마칠 수 있도록 하였다.

## 2. 연구 결과와 분석

본 연구의 실험처치에 따른 교육적 효과를 검증하기 위해 수집된 자료는 SPSS 25.0 프로그램을 사용하여 목적에 맞게 분석하였다.

소프트웨어 교육 효과성 검사, 컴퓨팅 사고력 검사, 사전·사후에 대하여 먼저 연구 집단이 정규성을 확보하고 있는지 확인하기 위하여 Shapiro-Wilks 검정을 실시하고 이에 따라 평균과 표준편차를 산출하였고 기술 통계를 산출하였다. 통계적 검증을 위한 유의수준은  $\alpha = .05$ 에서 살펴보았으며 그 효과에 대한 분석 결과는 다음과 같다.

### 1) 기술 통계 설계

유아 대상 소프트웨어 교육 사전과 사후의 교육 효과를 살펴보기 위해 유아 소프트웨어 교육 효과성과 컴퓨팅 사고력, 사전검사와 사후검사를 실시하였으며 본 연구의 설계를 도식화하여 제시하면 <표 IV-5>와 같다.

<표 IV-5> 실험 설계

대상	사전 검사	처치	사후 검사
만 5세 유아 41명	O <sub>1</sub>	X <sub>1</sub>	O <sub>2</sub>

X<sub>1</sub> : 유아 소프트웨어 교육 프로그램

O<sub>1</sub> : 사전 검사(유아 소프트웨어 교육 효과성, 컴퓨팅 사고력)

O<sub>2</sub> : 사후 검사(유아 소프트웨어 교육 효과성, 컴퓨팅 사고력)

2) 유아 소프트웨어 교육 효과성

(1) 정규성 검정

연구 집단이 정규성을 확보하고 있는지 확인하기 위하여 유아 소프트웨어 교육 효과성 사전 검사 결과를 이용하여 정규성 검정을 실시하였다.

사전 검사 결과 데이터를 바탕으로 후속 통계의 비모수/모수 통계 여부를 선정하기 위해 정규성 검정 방법으로 Shapiro-Wilks 검정을 실시하였고 결과를 살펴보면 <표 IV-6>과 같다.

<표 IV-6> 소프트웨어 교육 효과성 검사 정규성 검정

영역	세부영역	기술통계(N=41)				t	p
		M	SD	Max	Min		
가치	유용성	2.54	.925	2.83	2.24	.860	.000
	중요성	3.15	.910	3.43	2.86	.727	.000
	편리성	2.76	.830	3.02	2.49	.854	.000
	필요성	3.56	.808	3.82	3.31	.669	.000
	수업 만족도	3.39	.862	3.66	3.12	.750	.000
	진로 탐색	3.10	.889	3.38	2.82	.851	.000
태도	몰입도	2.80	1.005	3.12	2.49	.892	.001
	의사소통	3.12	.600	3.31	2.93	.696	.000
	협력 (공유, 협업)	2.88	1.229	3.27	2.49	.914	.005
	책임성	3.68	1.035	4.01	3.36	.866	.000
컴퓨팅 사고력 효능감	자기주도성	3.02	.935	3.32	2.73	.892	.001
	자신감	2.56	1.163	2.93	2.19	.875	.000
흥미	학습흥미	3.05	1.341	3.47	2.63	.903	.002

\* $p < .05$

유아 소프트웨어 교육 효과성 사전 검사에 대해 Shapiro-Wilks 정규성 검정을 실시한 결과, 유의도가 .05보다 낮게 나타나 정규성이 만족되지 않은 것으로 나

타났다. 정규성 확보에 따라 비모수적 통계방법을 사용하였다.

(2) 사전·사후 검사 결과 비교

본 연구를 통해 개발된 교육 프로그램이 유아 소프트웨어 교육 효과성을 살펴 보기 위해 사전 검사 결과를 비교한 후 정규성이 확보되지 못한 영역 전체를 Wilcoxon 부호 순위 검정을 실시하였다. <표 IV-7>의 소프트웨어 교육 효과성 사전·사후 검사 Wilcoxon 부호 순위 검정 결과를 살펴보면 다음과 같다. 유아 41 명을 대상으로 유아소프트웨어 교육 평가 사전·사후 검사를 실시한 결과, 전체영역에서 사전 검사 점수와 비교 했을 때 사후 검사 점수가 유의하게 증가했음을 알 수 있다. 즉, 유의수준 .05에서 모두 유의미한 차이가 있는 것으로 분석되었다.

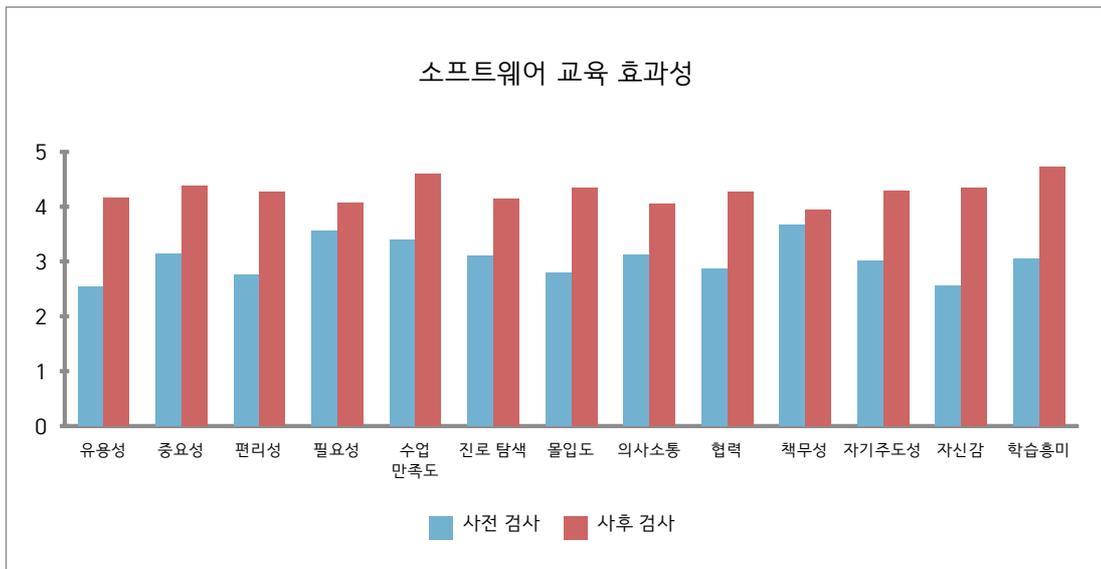
<표 IV-7> 소프트웨어 교육 효과성 검사 결과(Wilcoxon 부호 순위 검정)

영역	세부영역	N	사전 검사		사후 검사		z	p
			M	SD	M	SD		
가치	유용성	41	2.54	.925	4.17	.863	-5.679	.000
	중요성	41	3.15	.910	4.39	.997	-5.385	.000
	편리성	41	2.76	.830	4.27	.449	-5.344	.000
	필요성	41	3.56	.808	4.07	.985	-2.384	.017
	수업 만족도	41	3.39	.862	4.61	.862	-4.604	.000
	진로 탐색	41	3.10	.889	4.15	.910	-4.407	.000
태도	몰입도	41	2.80	1.005	4.34	.728	-5.165	.000
	의사소통	41	3.12	.600	4.05	1.203	-4.608	.000
	협력 (공유, 협업)	41	2.88	1.229	4.27	.923	-4.525	.000
	책임성	41	3.68	1.035	3.95	.893	-3.317	.001
컴퓨팅 사고력 효능감	자기주도성	41	3.02	.935	4.29	1.123	-4.514	.000
	자신감	41	2.56	1.163	4.34	.762	-5.265	.000
흥미	학습흥미	41	3.05	1.341	4.73	.742	-4.704	.000

\* $p < .05$

(3) 유아 소프트웨어 교육 효과성 검사 결과 분석

본 연구의 소프트웨어 교육 효과성 검사 결과 사전·사후의 차이 비교를 그래프로 나타내면 [그림 IV-5]와 같다. 차이 검증 결과를 살펴보면 전체적으로 사전에 비해 사후 결과가 향상된 것으로 볼 수 있다. 그러나 ‘나는 소프트웨어가 우리 생활에 필요하다고 생각한다.’라는 문항의 필요성은 사전 검사(M=3.56, SD=.808), 사후 검사(M=4.07, SD=.985)로 다른 문항에 비해 큰 차이가 없는 것으로 나타났다. ‘나는 다른 사람의 생각을 내 생각인 것처럼 말하면 안 된다는 것을 안다.’라는 문항의 책무성은 또한 사전 검사(M=3.68, SD=1.035), 사후 검사(M=3.95, SD=.893)로 큰 차이를 보이지 않는다. 만 5세 유아지만 소프트웨어 교육의 필요성과 프로그래밍 윤리에 대한 책임에 대해서도 알고 있는 것으로 해석할 수 있다. 이와 같은 결과는 유아 소프트웨어 교육이 효과가 있음을 의미한다.



[그림 IV-5] 소프트웨어 교육 효과성 검사 결과 비교

소프트웨어 교육 효과성 검사의 하위 요소인 영역별 사전·사후 총점을 비교해보면 <표 IV-8>과 같다. 가치, 태도, 컴퓨팅 사고력 효능감, 흥미 각 영역별에서 통계적으로 유의미하게 차이가 있는 것으로 나타났다. 이러한 결과는 유아가 활동을 통한 다양한 경험의 과정에서 얻어진 결과로 해석할 수 있다.

<표 IV-8> 소프트웨어 교육 효과성 검사 영역별 결과(Wilcoxon 부호 순위 검정)

영역	N	사전 검사		사후 검사		z	p
		M	SD	M	SD		
가치	41	3.08	.931	4.28	.874	-11.286	.000
태도	41	3.12	1.044	4.15	.957	-8.700	.000
컴퓨팅 사고력 효능감	41	2.79	1.074	4.32	.954	-6.934	.000
흥미	41	3.05	1.314	4.73	.742	-4.704	.000

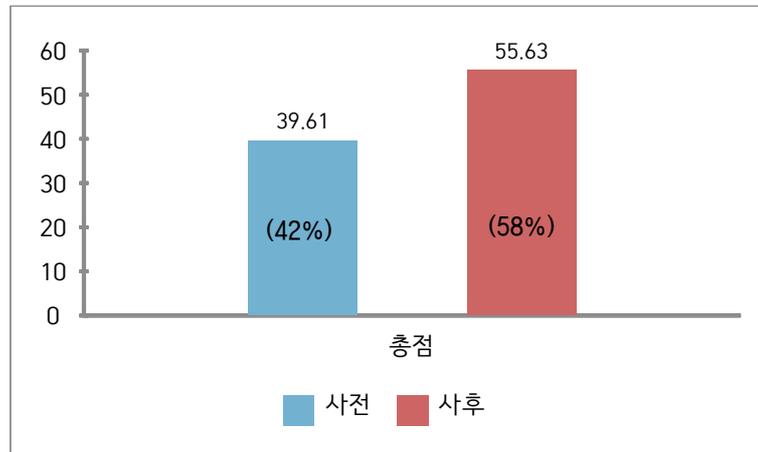
\* $p < .05$

소프트웨어 교육 효과성 검사의 사전·사후 총점을 비교해 보면 <표 IV-9>와 같다. 소프트웨어 교육 효과의 총점은 사전(M=39.61, SD=12.532)이 사후(M=55.63, SD=11.44)에 비해 높았으며 이러한 차이는 통계적으로 유의한 것으로 나타났다. 사전·사후 검사의 차이 비교를 그래프로 나타내면 [그림 IV-6]과 같다.

<표 IV-9> 소프트웨어 교육 효과성 검사 총점

영역	세부영역	N	사전 검사		사후 검사		z	p
			M	SD	M	SD		
소프트웨어 교육 효과성 총점		41	39.61	12.532	55.63	11.44	-16.456	.000

\* $p < .05$



[그림 IV-6] 소프트웨어 교육 효과성 검사 총점 비교

이와 같은 결과를 통해 유아 소프트웨어 교육 효과성에 있어 긍정적인 영향을 미치고 있음을 알 수 있다. 유아들은 매 활동 마다 문제를 발견하고 문제를 인식하는 과정을 거쳤을 것이다. 이러한 과정들이 호기심으로 이어지고 이후의 활동에 적극적으로 참여할 수 있는 동기부여가 되었을 것이다. 소집단 활동에서 또래 친구들과 함께 생각하고 앞선 친구의 활동을 지켜보고 확장해 나가면서 의사소통 능력과 협동 능력이 향상되었을 것이다.

### 3) 유아 소프트웨어 교육이 컴퓨팅 사고력에 미치는 효과

#### (1) 정규성 검정

연구 집단이 정규성을 확보하고 있는지 확인하기 위하여 컴퓨팅 사고력 사전 검사 결과를 이용하여 정규성 검정을 실시하였다.

사전 검사 결과 데이터를 바탕으로 후속 통계의 비모수/모수 통계 여부를 선정하기 위해 정규성 검정 방법으로 Shapiro-Wilks 검정을 실시하였고 결과를 살펴보면 <표 IV-10>과 같다.

컴퓨팅 사고력 사전 검사에 대해 Shapiro-Wilks 정규성 검정을 실시한 결과, 유의도가 .05보다 낮게 나타나 정규성이 만족되지 않은 것으로 나타났다. 정규성 확보에 따라 비모수적 통계방법을 사용하였다.

<표 IV-10> 컴퓨팅 사고력 검사 정규성 검정

CT 요소	컴퓨터과학 영역	기술통계(N=41)				t	p
		M	SD	Max	Min		
자동화	알고리즘 및 프로그래밍	2.88	1.552	3.37	2.39	.693	.000
알고리즘 시뮬레이션	알고리즘 및 프로그래밍	3.54	.840	3.80	3.27	.588	.000
문제 분석 문제 분해 시뮬레이션	알고리즘 및 프로그래밍	3.22	1.605	3.73	2.71	.485	.000
추상화 자동화 시뮬레이션	컴퓨팅 시스템 구성 및 동작원리	2.41	1.284	2.82	2.01	.890	.001
문제 분석 추상화	자료 분석 및 자료표현	2.24	1.625	2.76	1.73	.791	.000
자동화 알고리즘 시뮬레이션	알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리	1.71	1.647	2.23	1.19	.783	.000
문제 분석 문제 분해	자료 분석 및 자료표현	2.44	1.975	3.06	1.82	.619	.000
알고리즘	알고리즘 및 프로그래밍	3.71	1.055	4.04	3.37	.288	.000
알고리즘 시뮬레이션	알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리	1.98	1.275	2.38	1.57	.909	.003
추상화	자료 분석 및 자료표현	2.15	2.019	2.78	1.51	.635	.000

\* $p < .05$

## (2) 사전·사후 검사 결과 비교

본 연구를 통해 개발된 교육 프로그램이 유아 소프트웨어 교육에 미친 영향을 살펴보기 위해 사전 검사 결과를 비교한 후 정규성이 확보되지 못한 영역 전체를 Wilcoxon 부호 순위 검정을 실시하였다. <표 IV-11>의 컴퓨팅 사고력 검사 사전·사후 Wilcoxon 부호 순위 검정 결과를 살펴보면 다음과 같다. 유아 41명을 대상으로 컴퓨팅 사고력 사전·사후 검사를 실시한 결과, 두 개의 문항을 제외한 나머지에서 사후 검사가 유의미하게 증가했음을 알 수 있다. 유의수준 .05에서 유의미한 차이가 있는 것으로 분석되었다.

<표 IV-11> 컴퓨팅 사고력 결과 (Wilcoxon 부호 순위 검정)

CT 요소	컴퓨터과학 영역	N	사전 검사		사후 검사		z	p
			M	SD	M	SD		
자동화	알고리즘 및 프로그래밍	41	2.88	1.552	3.41	1.204	-2.282	.022*
알고리즘 시뮬레이션	알고리즘 및 프로그래밍	41	3.54	.840	3.83	.587	-3.207	.001*
문제 분석 문제 분해 시뮬레이션	알고리즘 및 프로그래밍	41	3.22	1.605	3.61	1.202	-2.000	0.46*
추상화 자동화 시뮬레이션	컴퓨팅 시스템 구성 및 동작원리	41	2.41	1.284	3.46	.925	-4.849	.000*
문제 분석 추상화	자료 분석 및 자료표현	41	2.24	1.625	2.63	1.699	-1.597	.110
자동화 알고리즘 시뮬레이션	알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리	41	1.71	1.647	3.37	1.220	-3.397	.001*
문제 분석 문제 분해	자료 분석 및 자료표현	41	2.44	1.975	2.83	1.843	-2.000	.046*
알고리즘	알고리즘 및 프로그래밍	41	3.71	1.055	3.80	.872	-1.000	.317
알고리즘 시뮬레이션	알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리	41	1.98	1.275	3.15	1.195	-5.261	.000*
추상화	자료 분석 및 자료표현	41	2.15	2.019	2.73	1.884	-2.449	.014*

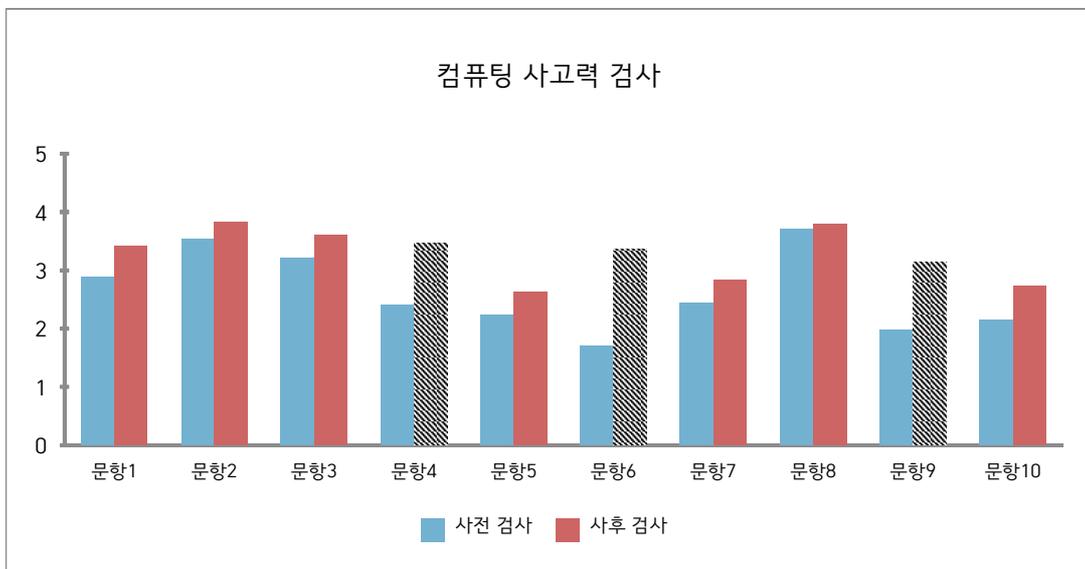
\* $p < .05$

### (3) 컴퓨팅 사고력 검사 결과 분석

본 연구의 컴퓨팅 사고력 향상 사전·사후의 차이 비교를 그래프로 나타내면 [그림 IV-7]과 같다. 차이 검증 결과를 살펴보면 5번 문항과 8번 문항을 제외하고 나머지 문항에서 사전에 비해 사후 결과가 향상된 것으로 볼 수 있다. ‘다른 친구와 물물교환을 통해 자신이 원하는 물건 얻는 5번 문항’에서는 사전 검사 (M=2.24, SD=1.625), 사후 검사(M=2.63, SD=1.699)로 유의미한 차이가 없는 것

으로 나타났다. ‘아이스크림을 순서대로 주문하는 방법에 대한 8번 문항’ 또한 사전 검사(M=1.055, SD=1.035), 사후 검사(M=3.80, SD=.872)로 유의미한 차이가 없는 것으로 나타났다.

이와 같은 결과는 유아 소프트웨어 교육이 유아들의 컴퓨팅 사고력 향상에 효과가 있음을 의미한다. 만 5세 유아 이지만 문제를 정확하게 이해하고 있고 충분히 풀 수 있는 걸로 해석할 수 있어 컴퓨팅 사고력 검사는 신뢰성이 있다는 것을 알 수 있다. 컴퓨터과학의 개념 중 컴퓨팅 시스템 구성 및 동작원리에 관련된 4번 문항과 알고리즘 및 프로그래밍, 컴퓨팅 시스템 구성 및 동작원리에 관련된 6번 문항, 9번 문항은 눈에 띄게 향상된 결과가 나타난 것을 살펴볼 수 있다.



[그림 IV-7] 컴퓨팅 사고력 검사 결과 비교

컴퓨팅 사고력 검사 컴퓨터과학 영역별 사전, 사후 차이를 비교해 보면 <표 IV-12>와 같다. 알고리즘 및 프로그래밍, 자료 분석 및 자료표현, 컴퓨팅 시스템 구성 및 동작원리가 사전에 비해 사후 결과가 향상된 것으로 볼 수 있다. 모든 하위요인에서 통계적으로 유의미하게 차이가 나타났다. 교육 활동을 통해 다양성을 확장하고 이러한 과정 속에 컴퓨팅 사고력 개념들도 자연스럽게 익힐 수 있게 된 것으로 볼 수 있다.

<표 IV-12> 컴퓨팅 사고력 컴퓨터과학 영역별 결과(Wilcoxon 부호 순위 검정)

컴퓨터과학 영역	N	사전 검사		사후 검사		z	p
		M	SD	M	SD		
알고리즘 및 프로그래밍	41	3.34	1.331	3.66	1.005	-3.572	.000
자료 분석 및 자료표현	41	2.03	1.431	3.33	1.120	-6.639	.000
알고리즘 및 프로그래밍 컴퓨팅 시스템 구성 및 동작원리	41	2.28	1.870	2.73	1.797	-3.561	.000

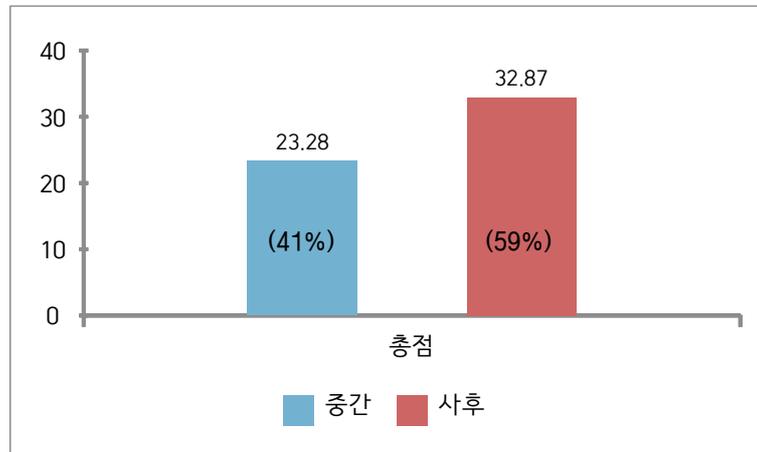
\* $p < .05$

컴퓨팅 사고력 검사 사전, 사후 총점을 비교해 보면 <표 IV-13>과 같다. 컴퓨팅 사고력의 총점은 사전(M=26.28, SD=14.877)이 사후((M=32.87, SD=12.631)에 비해 높았으며 이러한 차이는 통계적으로 유의한 것으로 나타났다. 사전·사후 검사의 차이 비교를 그래프로 나타내면 [그림 IV-8]과 같다. 이와 같은 결과를 통해, 유아 소프트웨어 교육이 컴퓨팅 사고력 향상에 긍정적인 영향을 미치고 있음을 알 수 있다.

<표 IV-13> 컴퓨팅 사고력 총점

CT 요소	컴퓨터과학 영역	N	사전 검사		사후 검사		z	p
			M	SD	M	SD		
컴퓨팅 사고력 총점		41	26.28	14.877	32.87	12.631	-7.445	.000

\* $p < .05$



[그림 IV-8] 컴퓨팅 사고력 검사 총점 비교

컴퓨팅 사고력과 같은 인지능력은 각국의 환경적 요인이나 생물학적 요인에 따라 다른 성과를 보일 수 있으므로(문용린, 유경재, 2007; Gardner, 1983) 국내 유아를 대상으로 한 실증적 연구가 이루어질 필요성이 있다. 또한 소프트웨어 교육성과는 기능, 기술을 활용하는 교육성과가 아니라 컴퓨팅 사고력을 함양하기 위해 접근해야 함에도 불구하고 다양한 용어로 불리는 컴퓨팅 사고력에 대한 논의 없이 검증되고 있다. 따라서 제대로 된 소프트웨어 교육을 한 후 창의적, 논리적 검사를 통한 접근이 아니라 컴퓨팅 사고력 검사를 통한 검증이 필요하다.

#### 4) 그 외 교육적 효과

유아 소프트웨어 교육을 진행하며 변화를 관찰한 내용과 교육 후 담임교사와의 면담 내용을 자세히 살펴보면 다음과 같다.

1차시 수업 '디지털 컬러' 부분 활동 비밀번호 만들기에서 교육 전 본인의 집 공동현관 비밀번호와 도어락 비밀번호를 아무렇지 않게 큰 소리로 이야기 하였다. 그러나 활동 후 나만의 비밀번호를 만들어 보는 시간에 다른 사람이 못 보게 종이를 가리고 작성하거나 "비밀번호가 뭔지 알려줄 수 있니?" 라는 질문에 "안 돼요. 혼자만 알고 있어야 하는 거예요." 라고 대답하는 변화를 보였다. 친구들과 등 돌리고 앉아 '말로 하는 댓글 경험해보기' 활동을 통해 얼굴이 보이지 않고 해서 나쁜 말을 사용하면 안 되는 이유에 대해 이해를 하고 실천하였다.

스마트 기기를 활용한 프로그래밍 도구 경험하기에서는 기능들에 대한 설명을 자세하기 해 주지 않아도 처음 해보는 코딩이지만 친구들과 협업하여 이야기를 나누며 여러 번 시도해 결국 답을 찾아가는 과정을 거쳤다. 이는 유아들이 스마트 기기 조작을 두려워하지 않고 이미 경험이 있는 만큼 잘 사용하고 있었다. 이는 좋은 방향으로 사용하도록 교육이 필요하다는 것을 시사하고 있었다. 로봇에게 명령을 내리기 전 스택카드를 나열해보는 활동에서 왼쪽, 오른쪽 이동을 어려워했다. 또한 스마트 기기 활동과는 다르게 로봇이 마음대로 움직이지 않으면 어렵다, 힘들다는 말을 자주 했다. 각 조별로 돌아가며 한 번씩 로봇이 되어 친구가 명령을 내리는 활동으로 변경해 수업을 진행 후 다시 로봇으로 수업을 진행하였다. 활동 후 물건을 가지러 갈 때 친구에게 빨리 갈 수 있는 길을 찾아 본인에게 명령을 내려달라며 놀이처럼 활동을 이어가는 효과가 있었다.

그러나 선행학습 금지법으로 인해 글자를 적는 활동에서 적지 못하는 유아가 있거나 맞춤법을 물어보며 적는 시간이 오래 걸려, 관련 활동에서는 그림을 그리거나 사진, 이미지 등을 미리 출력해 사용하거나 번호를 적는 방법 등으로 변경이 필요해 보였다.

교육 후 담임교사들과의 면담 내용을 살펴보면 다음과 같다.

소프트웨어 교육은 컴퓨터가 꼭 있어야 가능한지 알았지만 활동 내용을 살펴보니 누리과정 활동과 비슷한 부분도 있고 일상생활에서 주로 경험할 수 있는 내용들로 구성되어 있었다. 교육 내용들은 소프트웨어 교육보다 일반 교육에 더 가까워 보인다. 소프트웨어 교육이 무엇인지 미리 설명을 들어 이해는 했지만 조금 더 로봇이나 스마트 기기를 이용한 교육에 초점을 맞추어 진행하는 것도 나쁘지 않을 것 같다. 스마트 패드를 활용하여 블록을 맞추는 교육내용은 아이들이 재미있어하고 조금 더 하고 싶다거나 언제 하는지 물어보는 아이들이 많았다.

개인정보에 관련된 내용과 온라인 칭찬 릴레이에 대한 내용은 반드시 필요하고 한 시간으로 끝나는 게 조금 아쉽기도 했다. 글을 읽을 수 있는 원생들끼리는 미디어에서 본 이야기들을 많이 하는데 영상에서 들은 말과 댓글에서 본 단어가 나쁜 내용인지 모르고 친구들과 사용하는 행동에 대한 좋은 교육 내용이다.

스마트폰을 사용해 본 사람 손들어보라고 했더니 빠짐없이 모든 아이가 손을 들었다. 서로 경쟁하듯 많은 시간을 사용한다고 친구들에게 자랑하는 모습까지

보였다. 이러한 환경에서 소프트웨어 교육이 꼭 필요하다는 생각이 들었고 스마트폰을 바르게 사용하는 방법과 게임을 많이 하면 안 되는 이유에 대한 교육이 필요하다. 개인정보 보호나 스마트폰 활용, 과의존 등의 내용들은 양육자와의 올바른 애착관계 형성을 위해서 학부모들에게도 필요하며 함께 하는 교육이 되면 더 좋을 것 같다.

## V. 결론 및 제언

본 연구에서는 만 5세 대상 유아들의 소프트웨어 교육을 위하여 개발된 교육 과정이 소프트웨어 교육 효과성과 컴퓨팅 사고력에 미치는 효과를 확인하였다. 본 연구를 통해 얻어진 결과들을 중심으로 결론을 내리면 다음과 같다.

첫째, 개발된 교육과정은 유아 소프트웨어 교육 효과에 긍정적인 영향을 미치는 것으로 확인되었다. 하위 영역을 살펴보면 유용성, 중요성, 편리성, 필요성, 수업 만족도, 진로 탐색, 몰입도, 의사소통, 협력(공유, 협업), 책무성, 자기주도성, 자신감, 학습흥미 13가지 모든 요소에서 사후 검사 결과가 사전 검사 결과에 비하여 유의미한 차이를 나타내었다. 본 연구에서 개발한 교육과정은 코딩 능력, 로봇 활용 교육을 강조하는 기존의 유아 소프트웨어 교육과는 달리 소프트웨어 교육의 본질적인 측면을 강조하였으며 실생활과 관련된 활동으로 유아들이 놀이로 인식하고 쉽게 소프트웨어 교육에 접근할 수 있도록 하였다.

둘째, 개발된 교육과정은 컴퓨팅 사고력 향상에 긍정적인 영향을 미치는 것으로 확인되었다. 문제 분석, 문제 분해, 추상화, 알고리즘, 자동화, 시뮬레이션으로 구성된 10문 중 2문항을 제외하고 사후 검사 결과가 사전 검사 결과에 비하여 유의미한 차이를 나타내었다. 유아들이 소프트웨어 교육 활동에 호기심과 흥미를 가지고 참여하는 과정 속에서 자연스럽게 컴퓨팅 사고력이 향상된 것으로 보인다. 더불어, 문제 상황들을 해결할 때 또래 친구들과 함께 생각하고 공유하며 협업을 하는 경험을 통해 컴퓨터과학의 개념을 이해하고 컴퓨팅 원리를 발견할 수 있었을 것이라고 예측해 볼 수 있다.

본 연구를 통하여 얻을 수 있는 시사점은 다음과 같다.

첫째, 코딩 능력, 로봇을 통한 흥미 위주로 끝나는 활용교육이 아니라 소프트웨어 교육의 목적과 필요성에 대해 충분한 내용을 담고 있는 교육과정이 필요하다. 기존 연구에서는 코딩, 게임, 로봇 등으로 구성된 활용 교육 중심의 교육 프로그램을 제시하였다. 본 연구에서는 한 단계 더 나아가 국내외 교육과정 비교 분석 및 선행연구 분석을 통해 교육의 목적을 정의하고 내용체계를 개발해 교육 프로

그램을 운영하고 효과를 검증하였다. 이러한 시도는 추후 관련 연구에서 수정·보완 되더라도 유아 소프트웨어 교육의 본질을 이해하는데 많은 도움이 될 것이다.

둘째, 소프트웨어 교육의 효과성 검사 도구와 컴퓨팅 사고력 검사 도구를 개발하고 그 효과를 확인하였다. 앞서 언급하였듯이 선행연구에서는 일반적으로 유아용 창의성 검사 도구를 사용하고 있었다. 유아 수준에 맞는 컴퓨팅 사고력 검사 도구를 개발하기에는 절차 등이 쉽지 않고 오랜 기간 걸리기 때문에 파악된다. 본 연구에서 개발한 검사 도구는 컴퓨팅 사고력과 창의성의 차이를 구분해 추후 연구의 목적에 맞게 수정·보완하여 사용할 수 있을 것이다.

본 연구의 결과를 기초로 후속 연구를 위해 다음과 같이 몇 가지 제언을 하고자 한다.

첫째, 유아교사들은 소프트웨어 교육에 관심이 많은 편이나 관련 내용과 방법을 모르는 경우가 대부분이다. 소프트웨어 교육은 단순히 코딩 능력, 로봇 활용, ICT 활용 교육이라는 인식 개선이 필요하며 더불어 유아 소프트웨어 교육 이전에 유아교사들을 위한 소프트웨어 교육이 선행되어야 할 필요성이 있다. 따라서 예비 유아교사 양성과정과 유아교사 교육연수 프로그램이 개발될 필요가 있다.

둘째, 유아 소프트웨어 교육은 교육기관 내 활동으로 그치지 않고 부모의 참여와 지원이 필요하다. 가정과의 연계 교육 또는 부모 참여 교육을 통해 공감대를 형성하고 함께하는 교육이 되어야 한다. 이러한 교육은 소프트웨어 교육에서 더 나아가 정보윤리 교육과 더불어 스마트폰 과의존, 과몰입에 선한 영향력을 줄 수 있을 것이다.

셋째, 본 연구를 통해 소프트웨어 교육의 효과성과 컴퓨팅 사고력 향상에 긍정적인 효과를 미치는 것을 확인할 수 있었다. 그러나 이는 사전·사후 검사 결과를 통계처리 하여 확인한 결과로 유아 개개인에 대하여 어떠한 효과를 미쳤는지에 대해 살펴보는 것에 한계가 있었다. 따라서 후속 연구로 유아 개개인의 특성에 대하여 이해를 바탕으로 한 소프트웨어 교육 검사 도구 개발을 통해 살펴볼 수 있는 질적 연구를 제안한다.

넷째, 본 연구는 유치원 일부 교육기관을 대상으로 프로그램을 적용하였으므로, 이 연구를 전체 유아들의 결과로 일반화하기에 어려움이 있다. 따라서 각 교육기관들의 교육적, 지역적, 형태적 특성을 고려하여 본 소프트웨어 프로그램의 적용

및 효과를 규명하는 후속 연구가 필요하다고 본다.

인공지능 시대를 살아갈 아이들은 일상생활 속에서 또는 관련 직업을 통해 로봇과 때로는 협업하고 때로는 경쟁하는 과정을 겪을 것이다. 그 사회를 살아나갈 아이들이 미래사회 주인이기 때문에 어떠한 역량을 갖추게 되느냐에 따라 미래 사회의 동력이 될 수밖에 없다. 4차 산업혁명, 디지털 혁명, 정보 혁명, 소프트웨어 혁명, 인공지능 혁명 등 수많은 이름으로 새로운 시대를 정의하고 있지만 그 시대를 살아갈 대부분의 사람들이 해결해야 할 문제가 변하고 그에 따라 필요한 능력도 바뀌고 있다. 이러한 미래 사회에서 문제해결력과 컴퓨팅 사고력은 필수적일 수밖에 없다. 컴퓨팅 사고력은 컴퓨터과학 분야 뿐 아니라 일상생활의 다양한 문제를 효과적으로 해결하는 방법을 찾는 데에도 필요하다. 창의성을 바탕으로 아이디어와 상상력을 소프트웨어를 통해 실현하고 문제점을 소프트웨어로 해결하는 것이 일상화되는 사회에 컴퓨팅 사고력은 유아 뿐 아니라 모든 사람에게 필요한 역량이다.

다양한 변화의 시대를 준비하기 위해 세계 각국에서 좀 더 어린 나이부터 소프트웨어 교육에 관한 연구와 시도가 활발하게 이루어지고 있는 반면 국내에서는 유아 소프트웨어 교육이 제대로 자리 잡지 못하고 있는 실정이다. 소프트웨어 교육은 유아들이 이를 통해 일반적으로 갖추어야 할 기초지식과 보편적으로 길러줄 수 있는 능력을 위해 보편교육이 되어야 한다. 사교육을 통해서 선택적으로 이루어지는 교육이 아니라 국가 교육과정 속에 모든 유아들이 경험을 할 수 있는 시스템이 구축이 되어야 한다. 본 논문은 만 5세 대상 유아 소프트웨어 교육 과정을 통해 컴퓨팅 사고력 향상이라는 소프트웨어 교육의 궁극적인 목적 달성에 기여하고자 유아 소프트웨어 교육과정을 개발하였다. 이를 통해 유아 소프트웨어 교육에 대한 다양한 연구와 관심이 많아져 국가 수준의 교육과정으로 자리 잡을 수 있도록 확장을 기대한다.

## 참 고 문 헌

- 고병오 (2016). 아두이노를 기반으로 한 STEAM 교육에 관한 연구. 공주교육대학교 초등교육연구원. 53(4), 1-18.
- 고영희, 윤지영, 이루다, 이성국, 이승미, 정영찬, 허숙 (2018). 평가의 재발견. 서울: 맘에 드림.
- 과학기술정보통신부, 한국정보화진흥원 (2016). 2016년도 인터넷 과의존 실태조사 결과.
- 과학기술정보통신부, 한국정보화진흥원 (2018). 2018년도 스마트폰 과의존 실태조사 결과.
- 교육부, 보건복지부 (2015). 3-5세 연령별 누리과정: 해설서. 교육부 고시 제 2015-61호.
- 교육부 (2013). 5세 누리과정 교사용 지도서 11종-유치원/어린이집과 친구, 나와 가족, 우리 동네, 동식물과 자연, 건강과 안전, 생활도구, 교통기관, 우리나라, 세계 여러 나라, 환경과 생활, 봄·여름·가을·겨울.
- 교육부 (2015a). 초등학교 교육과정. 2018-162호. 교육부.
- 교육부 (2015b). 소프트웨어 교육 운영 지침. 교육부.
- 교육부 (2018a). 2015 개정 교육과정. 교육부.
- 교육부 (2018b). 유아교육법. 유아교육정책과.
- 김갑수 (2016). 미국 영국 독일 컴퓨터 교육과정에서 한국 컴퓨터 교육의 시사점. 한국정보교육학회논문지. 20(4), 421-432.
- 김대욱 (2019). 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 개념과 전략. 국제문화기술진흥원학술지. 5(1), 297-303.
- 김민자, 김정숙(2015). 컴퓨터 과학 : 학교 교육과정. 한국과학창의재단, 미래창조과학부.
- 김민자, 유길상, 김현철 (2016). 비전공자 프로그래밍 수업 창의적 산출물의 컴퓨팅 사고력기반 평가 루브릭 개발. 한국컴퓨터교육학회논문지. 20(2), 1-12.

- 김민정, 이원규, 김자미 (2017). 컴퓨팅사고력 측정에 사용되고 있는 도구 분석을 통한 새로운 검사도구 개발방향 제시. 한국컴퓨터교육학회논문지. 20(6), 17-25.
- 김상언, 김상희 (2018). 유아의 컴퓨팅 사고력을 위한 로봇활용교육 프로그램 개발 및 효과. 미래유아교육학회지. 26(1), 209-229.
- 김영희, 유경애 (2008). 유아교육과정. 서울: 동문사.
- 김자미, 이원규 (2014). 브루너의 이론에 근거한 인도의 정보교육과정 고찰. 한국컴퓨터교육학회논문지. 17(6), 59-69.
- 김재휘, 김동호 (2016). 컴퓨팅 사고력 향상을 위한 초등 피지컬 컴퓨팅 교육과정 개발. 한국정보교육학회논문지. 20(1), 69-82.
- 김정민, 홍일경, 김경민 (2016). 놀이를 통한 컴퓨팅 사고력 개발을 위한 유아 교육 개발에 관한 연구. 한국컴퓨터교육학회 학술발표대회논문집. 20(2), 187-190.
- 김철, 김갑수, 김현배, 정영식, 정인기 (2016). 초등 소프트웨어 교육과정 표준 모델 연구. 한국정보교육학회.
- 김혜림 (2007). 유아 컴퓨터 리터러시 교육 프로그램 개발 및 적용 효과. 박사학위 논문, 서울여자대학교 대학원.
- 김현정 (2002). 우리나라 유아의 일상생활에 대한 연구. 박사학위 논문, 이화여자대학교 대학원.
- 김현철(2015). 영국의 소프트웨어 교육 동향 및 한국교육에 주는 시사점. 서울: 한국교육개발원.
- 김형숙 (2015). 소프트웨어 교육을 위한 초등학교 교사들의 정보 교육 인식 및 개선방안. 석사학위 논문, 서울교육대학교 교육전문대학원.
- 김홍래, 이승진(2013). 외국의 정보(컴퓨터) 교육과정 현황 분석. 한국교육학술정보원(KERIS).
- 노운서 (2010). 만5세를 위한 컴퓨터 관련 교육활동 개발 및 그 효과에 관한 연구. 어린이미디어연구. 9(1), 63-87.
- 배영권, 신승기 (2017). 북유럽 국가의 교육체제 분석을 통한 소프트웨어 교육 발전방향 고찰: 에스토니아, 스웨덴, 덴마크, 노르웨이, 핀란드를 중심으로. 한

- 국교육개발원.
- 나원영 (2019). 언플러그드 컴퓨팅을 활용한 STEAM 교육이 초등학생의 창의적 문제해결력에 미치는 영향. 석사학위 논문, 경인교육대학교 교육전문대학원.
- 남궁선혜 (2002). 유아교육을 위한 미디어로서 로봇의 교육적 활용에 대한 연구. 열린유아교육연구. 15(3), 25-42.
- 박남수 (2018). 게임기반 유아 소프트웨어 교육에서 컴퓨팅 사고력에 영향을 미치는 요인규명. 박사학위 논문, 이화여자대학교 대학원.
- 박수연 (2012). 영아와 부모의 특성 및 육아유형과 영아발달 간의 관계: 월령단계 별 및 종단적 분석. 박사학위 논문, 이화여자대학교 대학원.
- 박주연, 김종혜, 김석희, 이현숙, 김수환 (2017). 초·중등 SW교육의 평가요소 개발. 한국컴퓨터교육학회논문지, 20(6), 47-59.
- 박창현 (2012). 유아교육과정 운영과 영향 변인간의 구조적 관계. 박사학위 논문, 중앙대학교 대학원.
- 박관우, 신승기 (2019). 초등학교 소프트웨어교육의 교육과정 개선을 위한 내용체계 및 교과 편성의 설계에 대한 연구. 정보교육학회논문지, 23(3), 273-282.
- 서영상 (2017). 유아 창의력 향상을 위한 색채블록 코딩 시스템. 박사학위 논문, 순천대학교 대학원.
- 심숙영 (2003). 유아 발달에 적합한 ICT 활용. 서울: 양서원.
- 신승기, 배영권 (2015). 핀란드의 코딩기반 소프트웨어 교육에 대한 고찰. 한국정보교육학회논문지. 19(1), 127-138
- 신승기, 배영권 (2014). 인도의 초등학교 컴퓨터 교육에 대한 분석 및 시사점. 한국정보교육학회논문지. 18(4), 585-594.
- 안상진, 이영준 (2014). 컴퓨터교과교육; 외국의 컴퓨팅 교육과정 변화와 함의. 한국컴퓨터교육학회논문지. 18(1), 47-51.
- 안성훈, 이상현 (2019). 핵심 역량 중심 2015 개정 초·중학교 SW교육과정 분석. 한국창의정보문화학회논문지. 5(1), 63-70.
- 양재명, 김자미, 이원규, 윤일규, 서정희 (2017). 초·중등 소프트웨어(SW)교육 역량 진단 도구 개발 연구. 한국교육학술정보원.
- 양진희 (2019). 영유아의 스마트기기 이용 현황 및 부모의 인식. 석사학위 논문,

- 한국교통대학교 대학원.
- 오연주, 안은숙, 김혜숙, 이은상, 이경실, 고영주, 이명희 (2003). 제작 과정 중심의 교재·교구 이론과 실제. 서울: 창지사.
- 원서연, 최연철 (2019). 교육용 프로그래밍 언어를 활용한 유아 소프트웨어 교육 경험의 의미 탐색. 학습자중심교과교육연구. 19(6), 1165-1187.
- 유구종 (1998). 유아 교사와 유아를 위한 유아 컴퓨터 교육의 실제. 서울: 양서원.
- 육아정책연구소 (2010). 초등학교 취학연령 및 유아교육 체제 개편 연구. 교육과학기술부 정책연구과제.
- 육아정책연구소 (2012). 육아정책 Brief, 2. 서울: 육아정책연구소.
- 육아정책연구소 (2018). 저출산에 대응한 어린이집·유치원 공급체계 전환과 중장기 적정수준 연구. 육아정책연구소.
- 이경희, 고은현, 조정원 (2018a). 유아 소프트웨어 교육의 현황과 접근방향 제안. 한국컴퓨터교육학회 학술발표논문지. 22(1), 7-10.
- 이경희, 이영석, 고은현, 홍찬의, 조정원 (2018b). 유아 소프트웨어 교육 내용체계 구성을 위한 제안. 한국컴퓨터교육학회 학술발표논문지. 22(2), 15-18.
- 이경희, 이영석, 고은현, 홍찬의, 문은경, 이경희 (2019). 유아 소프트웨어 교육 효과성 검증을 위한 평가 도구 개발 제안. 한국컴퓨터교육학회 학술발표논문지. 23(2), 77-79.
- 이경화 (2006). 한국 4, 5세 유아의 창의적 능력, 창의적 성격, 영역 창의성의 발달적 특성 연구. 유아교육연구. 26(5), 191-207.
- 이수련 (2009). 협동학습에서 유아의 인지수준이 창의성에 미치는 영향. 유아교육연구. 29(1), 57-82.
- 이우리 (2017). 스마트로봇 알버트BT를 활용한 코딩교육이 만 5세 유아의 의사소통능력 및 창의성에 미치는 효과. 석사학위 논문, 충신대학교 교육대학원.
- 이영재, 김영식 (2016). 초등학생의 소프트웨어 교육을 위한 피지컬 컴퓨팅 교구 선정기준 개발. 한국컴퓨터교육학회 학술발표대회논문집. 20(2), 35-38.
- 이영재, 전형기, 김영식 (2017). 2015 개정 교육과정의 초등학교 소프트웨어 교육을 위한 피지컬 컴퓨팅 교구 선택 기준 개발 및 적용. 한국정보교육학회논문지. 21(4), 437-450.

- 이진우 (2019). 협동적 문제해결학습에 기반한 유아코딩로봇활동이 교사 학습지원 행동과 유아의 수학적 문제해결력에 미치는 영향. 석사학위 논문, 강릉원주대학교 대학원.
- 이정림, 도남희, 오유정(2013). 유아의 미디어 매체 노출실태 보호책. 육아정책연구 소 연구보고 2013-15.
- 이철현 (2016). 초등 SW교육 방향 탐색 및 모델 개발. 한국실과교육학회지. 28(4), 207-222.
- 임명숙 (2017). 스마트로봇을 활용한 소프트웨어 교육이 유아의 창의성 향상에 미치는 효과. 석사학위 논문, 숭실대학교 교육대학원.
- 장성모 (1998). 교과와 내용과 형식, 교육학대사전. 서울: 하우동설.
- 정민경 (2018). 언플러그드 컴퓨팅을 활용한 STEAM 활동이 유아의 창의성 및 문제해결력에 미치는 효과. 석사학위 논문, 경남과학기술대학교 산업복지대학원.
- 정봄마지 (2013). UCC 제작활동이 유아의 디지털 리터러시와 정보윤리의식에 미치는 영향. 석사학위 논문, 이화여자대학교 교육대학원.
- 정선희 (2010). 보육시설 유아반 교재·교구 기준안 마련을 위한 기초 연구. 석사학위 논문, 공주대학교 대학원.
- 정옥분 (2002). 아동발달의 이해. 서울: 학지사.
- 정유진 (2019). 유아의 컴퓨팅 사고를 위한 놀이중심 언플러그드 컴퓨팅 프로그램 개발. 아시아문화학술원. 10(3), 1189-1199.
- 정지현 (2017). 예비유아교사의 소프트웨어 교육에 대한 관심도 분석: 관심중심수용모형(CBAM)을 중심으로. 한국산학기술학회논문지. 18(7), 431-440.
- 제주특별자치도교육청 (2017). 2017년 유치원 교육과정 및 방과후과정 내실화계획.
- 조준오, 홍광표 (2017). 누리과정 기반 유아 소프트웨어 교육과정 구성방향 탐색. 학습자중심교과교육연구. 17(15). 567-583.
- 조준오, 박창현, 홍광표 (2017). 유아 소프트웨어교육에 대한 유아교사의 인식과 요구. 학습자중심교과교육연구. 17(3), 83-106.
- 조준오, 홍광표 (2019). 유아 스마트폰 중독과 문제행동 간의 관계에서 자기조절

- 력의 매개 및 조절효과. 학습자중심교과교육연구. 19(10), 945-972.
- 최일선, 조운주 (2010). 유아교육 평가의 이론과 실제. 서울: 창지사.
- 최정원, 이영준 (2014). 컴퓨팅 사고력 평가 방안 설계. 한국컴퓨터정보학회 학술 발표논문지. 22(2), 177-178.
- 허희옥, 서정희 (2017). 해외 사례 검토를 통한 국내 SW교육 교사교육의 발전 방안 탐색. 한국교육공학회. 34(3), 711-741
- 한지원 (2019). 프로그래밍언어를 활용한 유아 소프트웨어교육 실행과정에서 겪은 유아의 경험. 석사학위 논문, 한국교원대학교 대학원.
- 홍혜경 (2002). 유아교수매체로서의 인터넷 활용에 관한 이론적 탐색. 유아교육학 논문집. 6(2), 93-113.
- 황성신 (2019). 소프트웨어 교육 활동 과정에서 나타난 유아의 컴퓨팅 사고 경험. 석사학위 논문, 한국교원대학교 대학원.
- 황해익 (2009). 유아교육평가의 이해. 경기: 정민사.
- KOTRA (2015). 한국인이 열광할 12가지 트렌드(KOTRA 전 세계 주재원이 취재한 세계의 지금 그리고 기회). 서울: 알키.
- Anja, B., Katja, E. (2015). Computing our future - Computer programming and coding Priorities, school curricula and initiatives across Europe. European Schoolnet.
- Bebras Challenge (2018). <http://bebras.org>.
- Berry, M. (2013). Computing in the national curriculum: A guide for primary teachers.
- Berry, M. (2014). Switched on Computing Second Edition Key Stage 1. Rising Stars UK Ltd.
- 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, 1, pp.25.
- Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. Journal of educational psychology, 76(6),

1051-1058.

Computer Science Teachers Association(CSTA) and the International Society for Technology in Education(ISTE) (2011). Computational Thinking Teacher Resources. Second Edition.

Cronbach, L. J. (1984). Essentials of psychological testing(4th ed.). New York: Harper & Row.

CSTA (2017). K-12 Computer Science Standards.

Devlin, K. (2003). Why universities require computer science students to take math. Communications of the ACM, 46(9), 37-39.

Ellis, P. T. (1965). Scientific Views of Creativity and Factors Affecting its Growth. Daedalus, The MIT Press on behalf of American Academy of Arts & Sciences, 94(3), 663-681.

England Department for Education (2015). Computing programmes of study: key stages 1 and 2. National curriculum in England.

European Commission (2015). Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. Draft 2015 Joint Report of the Council and the Commission on the implementation of the Strategic framework for European cooperation in education and training (ET2020). COM(2015) 408 final.

Gardner, H. (2007). 다중지능 (문용린, 유경재 역. 서울: 웅진출판사. - 원저: 1983년 출판)

Guilford, J. P. (1959). Three faces of intellect. American Psychologist, 14, 469-479.

Haugland, S. W. (2005). Technology in Early Childhood : Selecting or Upgrading Software and Web Sites in the Classroom. Early Childhood Education Journal, 32(5), 329-340.

Hazzan, O. (1999). Reducing abstraction level when learning abstract algebra concepts. Educational Studies in Mathematics, 40(1), 71-90.

- Helen Caldwell/Neil Smith (2017). Thiching Computing Unplugged in Primary Schools: Exploring primary computing through practical activities away from the computer. England: SAGE Publications, Inc.
- Hyson, C., & Eyman, A. (1986). Approaches to computer literacy in early childhood teacher education programs. In J. L. Wright, D. D. Shade(Eds), young children: A ctice learners in technological age. Washington, DC: NAEYC.
- HITSA (2013). Programming at Schools and Hobby Clubs. Retrieved from <http://www.innovatsioonikeskus.ee/en/programming-schools-and-hobby-clubs>
- Hohmann, C. (1990). Young Children and Computers. Michigan : The High/Scope Press.
- Inspiring Young People in STEM: LUMA FINLAND programme launched (2014). LUMA Center FINLAND.
- International Computer and Information Literacy Study (2013). Assessment Framework.
- K - 12 Computer Science Framework (2016). Retrieved from <http://www.k12cs.org>.
- Karen, S. (2006). Using augmented reality games to teach 21st century skills. In Proceedings of the 37th ACM Technical Symposium in Computing Science Education. ACM.
- Kamii, C., DeVries, R. (1978). Physical knowledge in preschool education: Implications of Piaget's theories. Englewood Cliffs, NJ: Prentice-Hall.
- Kamii, C., DeVries R. (1993). Physical knowledge in preschool education: implication of piaget's theory. New York: Teachers College Press.
- Koodi2016 (2014). Koodi2016-ensiapua ohjelmoinnin opettamiseen peruskoulussa. Retrieved from [https://s3-eu-west-1.amazonaws.com/koodi2016/Koodi2016\\_LR.pdf](https://s3-eu-west-1.amazonaws.com/koodi2016/Koodi2016_LR.pdf)
- Kramer, J. (2007). Is abstraction the key to computing?. Communications of

- the ACM, 50(4), 36-42.
- Lawshe, C. H. (1975) A Quantitative Approach to Content Validity. *Personnel Psychology*. 28, 563-575.
- Mednick, S. (1962). The associative basis of the creative process. *Psychological Review*, 69(3), 220 - 232.
- Ministry of Education, Singapore (2017). Teaching and Learning Guidelines in the use of ICT in Pre-school centers. [https://www.nel.sg/nel/slot/u566/Resources/Downloadable/pdf/\(MOE\)ICT%20Guidelines\\_Final.pdf](https://www.nel.sg/nel/slot/u566/Resources/Downloadable/pdf/(MOE)ICT%20Guidelines_Final.pdf)
- Ministry of Education, Singapore (2012). Singapore kindergarten-curriculum-framework.
- NAEYC (1996). NAEYC Position Statement: Technology and young Children-ages three through eight. *Young Children*, 51(6), 11-16.
- Nevo, D., Elana, S. (1986). Evaluation Standards for the Assessment of Alternative Testing Methods: An Application. *Studies in Educational Evaluation*. 12, 149-158.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. NY: Basic Books.
- Papert, S. (1987). Information technology and education: Computer criticism vs. technocentric thinking. *Educational researcher*, 16(1), 22-30.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95-123.
- Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & Cognition*, Washington, 1-6.
- Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New pathways into robotics: Strategies for broadening participation. *Journal of*

- Science Education and Technology, 17(1), 59-69.
- Seiter, L., & Foreman, B., (2013). Modeling the learning progressions of computational thinking of primary grade students. In Proceedings of the ninth annual international ACM conference on International computing education research, 59-66. ACM.
- Siegle, D. (2004). The merging of literacy and technology in the 21st century: a bonus for gifted education. *Gifted Child Today*, 27(2), 32-35.
- Sridhar, L., Farida, K., Sahana, M. (2013). *CMC: A Model Computer Science Curriculum for K-12 Schools*. 3rd Edition. Indian Institute of Technology Bombay, Mumbai.
- Tim B., Ian, H. (2006). *Computer Science Unplugged: An enrichment and extension programme for primary-aged children*. University of Canterbury.
- Torrance, E. P. (1978). Giftedness in solving future problems. *Journal of Creative Behavior*, 12(2), 75-86.
- Tyler, R. W. (1949). *Basic principles of curriculum and instruction*. Chicago: University of Chicago Press.
- Vygotsky, L. S. (2009). *Mind in Society*(Mind in Society: The Development of Higher Psychological Processes. (정희욱 역. 서울: 학이시습. - 원저: 1978년 출판).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational Thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, 366, 3717-3725.

<ABSTRACT>

# A Study on Curriculum Development for Early Childhood Software Education

**KyungHee Lee**

Major of Computer Education, Faculty of Science Education  
Graduate School, Jeju National University

**Supervised by professor Jungwon Cho**

The purpose of this study is to develop and apply early childhood software curriculum for 5 year olds. Through this, verifying the effectiveness of software education, and examining the effects on computational thinking improvement.

The AI era is no longer a distant future, but a reality world where software is the foundation and center of life. Society is changing rapidly, and with the change, the structure of life is also becoming diverse and complicated. Infants who will live in the future will be faced with a wider variety of problems to solve in future societies. Around the world, studies and efforts for starting software education have been conducted from an early age to develop computational thinking and problem solving skills. Software education should be a general education that everyone should learn for future life. The age of five is a period of rapid growth and change in all areas of

development. Even if they don't have a programmer or related job, software education is essential at that time. As early as possible, it's a good idea to start teaching to remind themselves how to think through experiences and play activities and to discover problems and find solutions by themselves. In order to contain such contents, it is necessary to approach not just end up in the coding ability development and robot utilization but indirect and natural activities composed of experience and various plays. In other words, through experience, there is a need for a systematic curriculum that can cultivate computational thinking, the ultimate goal of software education. In addition, not a selective education through private education, but a system in which all children can be educated within the national curriculum should be established.

In this study, in order to establish a system that can be started at the national level as a curriculum that can be linked to elementary and secondary schools in harmony with the national early childhood curriculum, an early childhood software curriculum for 5 years old was developed.

For this purpose, this study proceeded with the following procedure.

First, the domestic and foreign early childhood software curriculum, related research trends, literature analysis, and previous studies were investigated and compared. This analysis was the basis for the development of educational purpose and content system.

Second, by developing the curriculum consisting of educational software and contents system for early childhood software, and completed it through content validation test by expert verification, and Delphi survey.

Third, after applying the developed software curriculum to 5 years old children, the results were verified and analyzed by using the software education effectiveness test tool and the computational thinking test tool.

The outcomes of this process are early childhood software education objectives, content systems, teaching and learning methods, educational programs, and testing tools.

In this study, the following educational effects were verified.

First, the results of the software education effectiveness test showed that the curriculum developed in this study was effective for the education of early childhood software for 5 years old. Compared to before education experience, understanding, attitude, computational thinking efficacy, and interest in the value of software education, which is a sub-factor of test, were high and these showed a statistically significant difference.

Second, the results of the computational thinking test showed that the curriculum developed in this study was effective in improving the computational thinking ability of 5 years old children. Compared to before education experience, computational thinking improved after experiencing education, which was statistically significant.

The implications of this study are as follows.

First, this study presented the curriculum that can be linked to elementary and middle school, in harmony with the national level of early childhood curriculum, with sufficient contents about the purpose and necessity of software education. The educational program developed on this basis has proved to have sufficient educational effect. In future study in related fields, it may be helpful to understand the nature of early childhood software education and computational thinking.

Second, a software effectiveness test tool and computational thinking test tool for infant level were presented. Through this, it has demonstrated the effectiveness of early childhood software education and improved computational thinking. The computational thinking test needs to be used separately from the logic test and the creativity test. The test tool developed in this study can be modified and supplemented for future research purposes, and it will be helpful for understanding the meaning and necessity of the self-test of computational thinking.

It is expected that future research will continue to improve the curriculum

developed in this study and develop into linkage education with families, parental involvement education, in-service early childhood teacher education, and preparatory early childhood teacher education. In addition, if a quantitative computational thinking test tool with validity and reliability for 5 years old is developed, the effect of improving the computational thinking ability of individual infants can be analyzed.

Keywords : Early childhood software education, Software education, Computational thinking, Curriculum, Software education effect

## 부 록

<부록 1> 연구 안내서 및 연구 참여 동의서 .....	121
<부록 2> 소프트웨어 교육 효과성 검사지 .....	124
<부록 3> 컴퓨팅 사고력 검사지 .....	125
<부록 4> 유아 소프트웨어 교육 프로그램 .....	135

<부록 1> 연구 안내서 및 연구 참여 동의서

**연구 안내서 및 연구 참여 동의서**

학부모님께

안녕하세요.

4차 산업혁명 시대의 도래로 미래인재 양성을 목표로 한 소프트웨어 교육이 중요시 되고 있습니다. 유아의 컴퓨팅 사고력과 문제해결 능력을 길러주기 위한 소프트웨어 교육 프로그램을 개발하기 위해 아래와 같이 연구를 진행하고자 합니다. 본 연구의 결과는 향후 만 5세 소프트웨어 교육의 중요한 기초 자료가 될 것입니다.

본 연구의 참여 결정은 원생의 보호자 의사에 달려 있으니 참여를 원하지 않으면 참석시키지 않도록 하겠습니다. 이에 따른 불이익은 없습니다.

또한, 보호자가 참여를 원하더라도 원생이 참여를 원하지 않을 경우, 수업 참여 중 힘들어 할 경우 수업에서 제외하도록 하겠습니다.

본 연구를 통해 얻어진 개인정보(이름 등)및 검사 결과는 연구 외에 다른 용도로 사용되지 않습니다. 또한 연구자료 보관기간이 종료된 후에는 안전하게 폐기됩니다.

이 안내장은 학부모님들께 연구에 대한 정보를 제공하기 위해 제작 되었습니다. 아래의 정보를 읽어보시고 혹시 궁금한 점이 있으면 언제든지 담당 연구자에게 문의주시면 답변 드리겠습니다. 연구를 위해 도움을 주셔서 진심으로 감사드립니다.

- ▷ **연구제목** : 유아 소프트웨어 교육을 위한 교육과정 및 프로그램 개발
- ▷ **연구목적** : 만 5세 유아의 소프트웨어 교육 프로그램 개발을 위한 수업 및 평가
- ▷ **연구 대상자 및 인원** : 만 5세 유아 40명(예정)
- ▷ **수업 시간 및 기간** : 총 10차시(매일 1차시 30~50분 이내 수업)
- ▷ **연구 담당자** : 이경희(제주대학교 컴퓨터교육과) / 지도교수 : 조정원
- ▷ **연락처** : dreamer@jejunu.ac.kr 010-2509-\*\*\*\*

※ 본 연구 결과는 반드시 비밀이 보장되며, 연구 통계 외 다른 목적으로 일체 사용되지 않습니다.

### 연구 안내서 및 연구 참여 동의서(제출용)

▷ 본인은 위 연구의 목적, 연구대상자(유아)의 참여기간, 연구절차 및 방법에 대해 충분히 이해하였습니다.

동의합니다.

동의하지 않습니다.

▷ 본인은 이 연구과정에서 진행될 수 있는 평가를 위한 검사에 대해 충분히 이해하였습니다.

동의합니다.

동의하지 않습니다.

▷ 본인은 이 연구에 자유로운 의사에 따라 참여하며 필요할 경우 중간에 참여를 중단할 수 있다는 것을 이해하였습니다.

동의합니다.

동의하지 않습니다.

▷ 설문조사 참여 동의여부에 대한 서명란입니다.

유아이름 : \_\_\_\_\_ (학부모서명)

연구대상자와의 관계 : 유아의 \_\_\_\_\_

2019년 7월 일

\* 학부모님께서 보내드린 동의서에 서명하셔서 담임 선생님께 보내주시면 감사하겠습니다.

수업 일정 및 수업 내용(학부모 보관용)		
수업 일정	활동명	수업 내용(활동 목표)
1차시	나를 나타내는 것들이 있어요!	<ul style="list-style-type: none"> <li>· 글자와 숫자, 사진 등 나를 나타낼 수 있는 정보를 찾아본다.</li> <li>· 개인정보를 사용하는 정보기에 관심을 가진다.</li> <li>· 알파벳과 기호, 숫자를 조합하여 비밀번호를 구성해본다.</li> </ul>
2차시	온라인 칭찬 릴레이	<ul style="list-style-type: none"> <li>· 인터넷, 휴대폰 등의 정보기기를 사용할 때의 예절에 관심을 갖는다.</li> <li>· 우리 생활을 편리하게 돕는 정보기기를 알고 사용해 본다.</li> <li>· 친구의 좋은 점을 찾아서 짧은 문장으로 표현할 수 있다.</li> </ul>
3차시	이슬이의 심부름을 어떻게 도와줄까?	<ul style="list-style-type: none"> <li>· 심부름을 가는 길에 벌어질 수 있는 상황 예측해본다.</li> <li>· 문제를 해결하기 위해 세 가지 단계로 나눠서 생각해 보는 경험을 한다.</li> <li>· 같은 문제라도 해결할 수 있는 방법은 여러 가지가 있음을 안다.</li> </ul>
4차시	문어와 꿀벌 이야기	<ul style="list-style-type: none"> <li>· 해결하고자 하는 문제를 정확히 인식하고 해결할 수 있다.</li> <li>· 컴퓨터를 이용한 간단한 코딩프로그램을 경험해본다.</li> <li>· 문어의 생김새와 특징을 안다.</li> </ul>
5차시	어떤 생일파티를 하고 싶니?	<ul style="list-style-type: none"> <li>· 생일파티의 의미를 알고, 다양한 축하 방법을 생각할 수 있다.</li> <li>· 생일파티 준비 과정을 몇 가지 단계로 나누어 말해본다.</li> <li>· 생일파티 준비를 위하여 하고 싶은 일을 스스로 계획할 수 있다.</li> </ul>
6차시	나만의 김밥을 만드는 방법은?	<ul style="list-style-type: none"> <li>· 김밥에 들어갈 재료의 조화된 맛을 예측해본다.</li> <li>· 김밥을 만들 때 생길 수 있는 문제를 예측하고, 해결방법을 생각해본다.</li> <li>· 김밥의 요리 순서를 글과 그림으로 표현할 수 있다.</li> </ul>
7차시	순환운동 만들기 - 우리가 만든 한 바퀴 운동	<ul style="list-style-type: none"> <li>· 순서에 따라 활동하다가 벌어지는 문제를 친구와 협력하여 해결해본다.</li> <li>· 몇 가지 운동 동작을 선택하여 패턴을 만들 수 있다.</li> <li>· 신체 활동에 적극적으로 참여하며 즐거움을 느낀다.</li> <li>· 일상생활에서 자주 사용하던 도구의 장단점을 찾아서 말할 수 있다.</li> </ul>
8차시	생활 속에서 함께하는 로봇들 알아보기	<ul style="list-style-type: none"> <li>· 비슷한 기능을 가진 도구들의 특징 비교할 수 있다.</li> <li>· 자기 몸을 스스로 깨끗하게 하는 방법을 알고 실천한다.</li> </ul>
9차시	로봇에게 무엇을 명령할까?	<ul style="list-style-type: none"> <li>· 생활을 더 편리하게 하기 위해 로봇을 사용해 보는 경험을 한다.</li> <li>· 여러 가지 명령 카드 중 나에게 필요한 것을 선택해 사용할 수 있다.</li> <li>· 로봇에게 원하는 일을 명령하기 위해 순서 생각하며 코딩해본다.</li> </ul>
10차시	길을 찾아가는 걸 도와줘, 로봇!	<ul style="list-style-type: none"> <li>· 일상생활 속의 문제를 비판적으로 생각해 보는 경험을 한다.</li> <li>· 코딩한 로봇을 사용해 보고 잘못된 점 찾아서 수정할 수 있다.</li> <li>· 우리 생활을 더욱 편리하게 할 수 있는 방법을 창의적으로 생각해 본다.</li> </ul>

\* 매일 1차시 30~50분의 활동으로 진행됩니다.

## 유아 소프트웨어 교육 효과성 검사

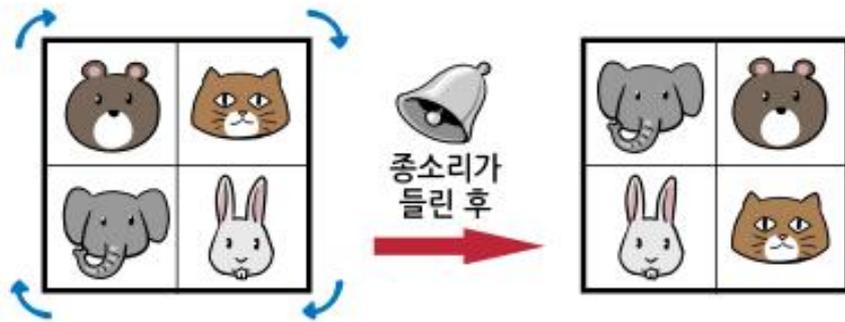
소프트웨어는 지난 2주 동안 선생님이랑 친구들이랑 같이 수업을 하면서 배운 내용이다. 어떠한 일들을 해결하기 위한 순서와 방법을 지시하는 명령들을 배웠고, 그 과정들을 위해서 어떤 것들이 필요한지를 함께 알아보았다. 소프트웨어에 대한 네 생각이 궁금해서 몇 가지 물어볼 거야. 정말 그렇다고 생각하면 5, 조금 그렇다고 생각하면 4, 보통이면 3, 그렇지 않으면 2, 전혀 그렇지 않으면 1에 V라고 표시하면 된다.

영역	세부영역	설문문항	평가요소	정말 그렇다	조금 그렇다	보통이다	그렇지 않다	전혀 그렇지 않다
				5	4	3	2	1
가치	우용성	1. 나는 SW가 문제를 해결할 때 도움이 된다고 생각한다.	SW 중요성, SW 교육의 유용성	5	4	3	2	1
	중요성	2. 나는 SW를 배우는 것이 중요하다고 생각하고 배우지 않는 친구들보다 함께 공부하고 싶다.	SW 교육 중요성	5	4	3	2	1
	편리성	3. 나는 SW가 우리 생활을 편리하게 해준다고 생각한다.	SW 편리성	5	4	3	2	1
태도	필요성	4. 나는 SW가 우리 생활에 필요하다고 생각한다.	SW 수업 필요성, SW 필요성	5	4	3	2	1
	수업 만족도	5. 나는 SW활동 시간을 좋아한다.	SW 수업 선호도, 수업 내용에 대한 만족도, 수업에 대한 기대감	5	4	3	2	1
	진로 탐색	6. 나는 학교에 가서도 SW를 더 많이 공부하고 싶다.	SW 관련 직업에 대한 인식, SW 관련 전공에 대한 인식, SW 교육에 대한 의지	5	4	3	2	1
	몰입도	7. 나는 SW활동을 할 때 집중을 잘한다.	프로그래밍 몰입	5	4	3	2	1
	의사소통	8. 나는 SW활동 시간에 친구들과 더욱 많이 이야기 나누는구나.	수업을 통한 의사소통, 친구와 상호작용, 교사와 상호작용	5	4	3	2	1
흥미	컴퓨팅 사고력 호기심 창의성	협력 (공유, 협업)	수업에서의 협동심, 타인과의 협력을 통한 창의성 발현, 공동체의식	5	4	3	2	1
		책무성	프로그래밍 윤리에 대한 책임	5	4	3	2	1
		자기주도성	과제를 해결하려는 노력, 자신의 생각을 표현하는 능력, 프로그래밍 실행에 대한 탐구심	5	4	3	2	1
흥미	학습흥미	자신감	적극적인 수업참여, 프로그래밍에의 자신감, 정보처리 사고과정의 자신감, SW 수업을 통한 성취감	5	4	3	2	1
		13. 나는 SW활동이 기다려지고 재미있다.	수업에 대한 흥미, 학습활동에 대한 흥미, 수업에 대한 지속의지, SW에 대한 관심	5	4	3	2	1

<부록 3> 컴퓨팅 사고력 검사지

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input checked="" type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input checked="" type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input checked="" type="checkbox"/> 자동화	<input type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

1. 동물 친구들이 모여 자리 바꾸기 게임을 하는 중입니다. 게임의 규칙은 종소리가 들릴 때 마다 화살표 방향으로 한 칸씩 옮겨 앉는 것입니다.



★ 질문:

종소리가 한 번 들렸습니다. 동물 친구들이 어디에 앉아있을지 숫자로 적어 봅시다.


①

②

③

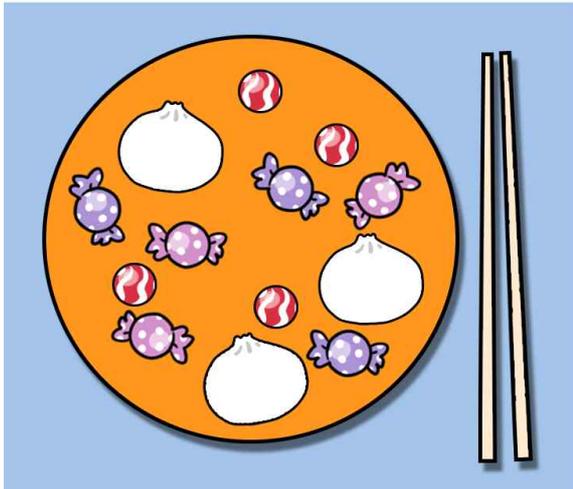
④

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input checked="" type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input checked="" type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input checked="" type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

2. 도훈이는 젓가락 사용하는 방법을 배우고 있습니다.

- 엄마가 정한 규칙 : 사탕은 꼭 손으로 먹어야 한다.

만두는 젓가락으로만 먹어야 한다.

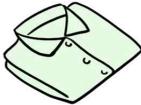
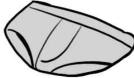


★ 질문:

도훈이가 젓가락으로 먹어야 하는 것을 골라 모두 동그라미 해보세요.

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input checked="" type="checkbox"/> 문제 분석	<input checked="" type="checkbox"/> 문제 분해	<input checked="" type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input checked="" type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

3. 아래 네모 안에는 은성의 옷입니다.

셔츠	스웨터	바지	속옷	양말	신발
					

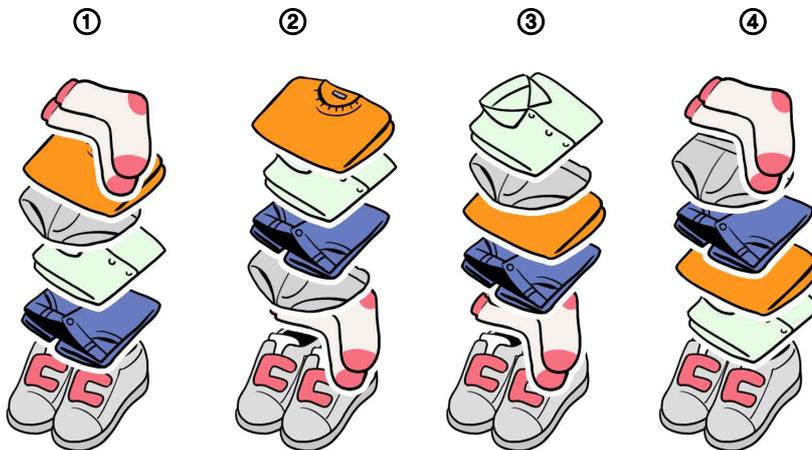
- 은성의 아빠가 은성이 옷을 정리하고 있습니다.



- 은성은 위에서부터 옷을 순서대로 입는다.
- 은성은 셔츠 위에 스웨터를 입고 유치원에 가고 싶어 한다.

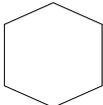
★ 질문:

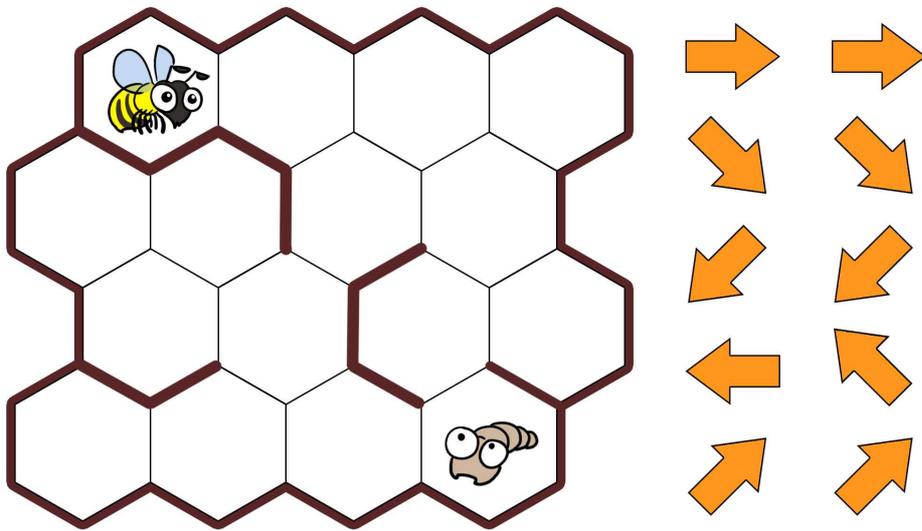
은성이 즐겁게 유치원에 갈 수 있도록 정리된 옷을 선택해 주세요.



컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input type="checkbox"/> 알고리즘 및 프로그래밍
<input checked="" type="checkbox"/> 추상화	<input type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input checked="" type="checkbox"/> 자동화	<input checked="" type="checkbox"/> 시뮬레이션	<input checked="" type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

4. 꿀벌은 애벌레에게 꿀을 전달해야 합니다.

- 꿀벌은 벌집 속  얇은 벽만 통과하여 움직일 수 있습니다.



★ 질문:

화살표를 이용하여 벌이 애벌레에게 꿀을 전달할 수 있게 도와주세요.



주의사항: 벌은 5번의 이동으로 애벌레에게 도착해야 합니다.

4번째 이동 경로는 이미 정해져 있습니다.

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input checked="" type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input type="checkbox"/> 알고리즘 및 프로그래밍
<input checked="" type="checkbox"/> 추상화	<input type="checkbox"/> 알고리즘	<input checked="" type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

5. 친구들과 물물교환을 통해 원하는 것을 가질 수 있습니다.

- 도훈이는 지금 풍선을 가지고 있고, 사탕으로 바꾸고 싶어 합니다.
- 채은이는 지금 사탕을 가지고 있고, 아이스크림으로 바꾸고 싶어 합니다.
- 수린이는 아이스크림을 가지고 있습니다.



★ 질문:

어떤 물물 교환 과정을 거쳐 도훈이는 사탕을 가질 수 있을까요?

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input checked="" type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input checked="" type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input checked="" type="checkbox"/> 자동화	<input checked="" type="checkbox"/> 시뮬레이션	<input checked="" type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

6. 친구들이 모여 로봇게임을 하고 있습니다. 로봇의 이름은 신비인데, 신비는 직진, 오른쪽, 왼쪽의 세 가지 명령만 알아듣습니다.



- 신비에게 직진이라고 명령하면 건물이나, 울타리, 수풀 등에 부딪칠 때까지 앞으로 걸읍니다.
- 신비에게 오른쪽이라고 명령하면 오른쪽으로 방향은 바꾸지만 움직이지는 않습니다.
- 신비에게 '왼쪽'이라고 명령하면, 왼쪽으로 방향은 바꾸지만 움직이지는 않습니다.

★ 질문:

신비는 지금 수풀을 바라보고 있으며 왼쪽 아래 구석에 있습니다. 신비가 집으로 갈 수 있도록 명령을 내려 도와주세요.

- ↑ 직진 → 오른쪽 ↑ 직진 ← 왼쪽 ↑ 직진 → 오른쪽 ↑ 직진 ← 왼쪽 ↑ 직진
- 오른쪽 ↑ 직진 ← 왼쪽 ↑ 직진 ← 왼쪽 ↑ 직진
- 오른쪽 ↑ 직진 ← 왼쪽 ↑ 직진 → 오른쪽 ↑ 직진 → 오른쪽 ↑ 직진
- 오른쪽 ↑ 직진 ← 왼쪽 ↑ 직진 → 오른쪽 ↑ 직진 ← 왼쪽 ↑ 직진

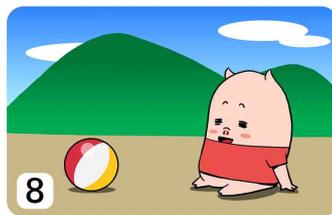
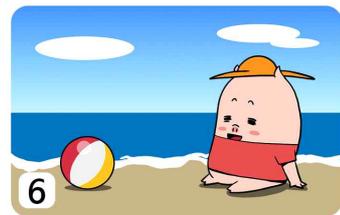
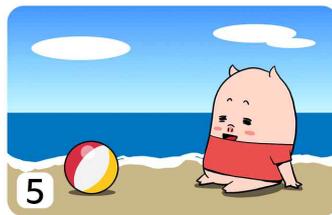
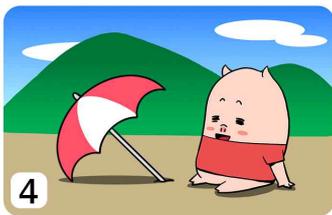
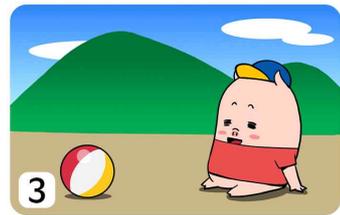
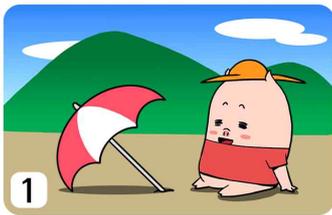
컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input checked="" type="checkbox"/> 문제 분석	<input checked="" type="checkbox"/> 문제 분해	<input type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input type="checkbox"/> 알고리즘	<input checked="" type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

7. 채은이는 8장의 사진을 가지고 있습니다. 채은이는 사진 들 중 한 장을 민경이에게 주고 싶어 합니다. 민경이에게 세 가지 질문을 해 원하는 사진을 주려고 합니다.

채은 질문	민경 대답
파라솔이 나온 사진이 좋은가요?	네
머리에 모자를 쓰고 있는 사진이 좋은가요?	아니요
바다가 나온 사진이 좋은가요?	네

★ 질문:

다음 중 채은이가 민경이에게 줄 사진은 어떤 것일까요?



컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input checked="" type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input checked="" type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

8. 아이스크림 가게에서는 내가 원하는 맛의 아이스크림의 종류를 말하면 순서대로 쌓아줍니다.

★ 질문:

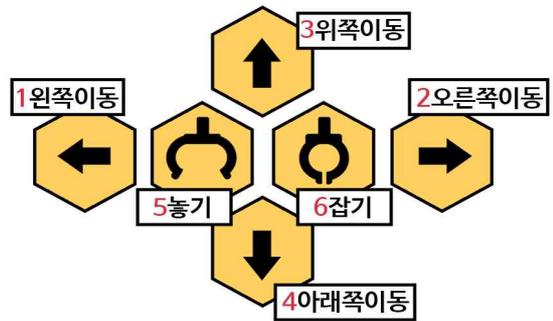
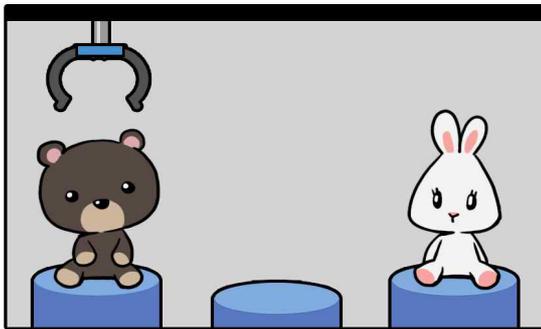
그림과 같은 모양의 아이스크림을 주문하기 위해서는 어떻게 이야기해야 할까요?



1. 저는 딸기맛-바닐라맛-초코맛으로 주세요.
2. 저는 초코맛-바닐라맛-딸기맛으로 주세요.
3. 저는 딸기맛-초코맛-바닐라맛으로 주세요.
4. 저는 초코맛-딸기맛-바닐라맛으로 주세요.

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input checked="" type="checkbox"/> 알고리즘 및 프로그래밍
<input type="checkbox"/> 추상화	<input checked="" type="checkbox"/> 알고리즘	<input type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input checked="" type="checkbox"/> 시뮬레이션	<input checked="" type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

9. 인형 뽑기 기계에는 6가지 작동방법이 있습니다. 기계 안에는 곰 인형과 토끼 인형이 있습니다.



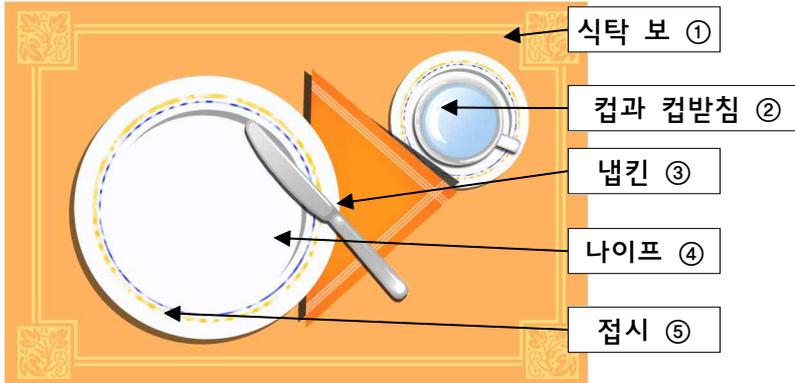
★ 질문:

명령 단추를 이용하여 곰 인형을 토끼 인형 옆으로 위치를 이동해 봅시다.

작동방법 그림에 적힌 숫자를 이용해서 작성해 주세요. 예) 왼쪽 이동:1, 위쪽 이동:3

컴퓨팅 사고력 검사지		
Computational Thinking Skills		컴퓨터 과학 영역
<input type="checkbox"/> 문제 분석	<input type="checkbox"/> 문제 분해	<input type="checkbox"/> 알고리즘 및 프로그래밍
<input checked="" type="checkbox"/> 추상화	<input type="checkbox"/> 알고리즘	<input checked="" type="checkbox"/> 자료분석 및 자료표현
<input type="checkbox"/> 자동화	<input type="checkbox"/> 시뮬레이션	<input type="checkbox"/> 컴퓨팅 시스템 구성 및 동작원리

10. 수린이는 부모님과 함께 식사 준비를 하고 있습니다.



★ 질문:

식탁에 어떤 순서로 물건들을 놓았을까요?

1. ①식탁 보→ ③냅킨→ ②컵과 컵받침→ ④ナイ프→ ⑤접시
2. ①식탁 보→ ③냅킨→ ②컵과 컵받침→ ⑤접시→ ④ナイ프
3. ③냅킨→ ④ナイ프→ ①식탁 보→ ②컵과 컵받침→ ④접시
4. ①식탁 보→ ②컵과 컵 받침→ ③냅킨→ ⑤접시→ ④ナイ프

<부록 4> 유아 소프트웨어 교육 프로그램

주요개념     추상화     알고리즘     프로그래밍

### 15. 문어와 꿀벌 이야기

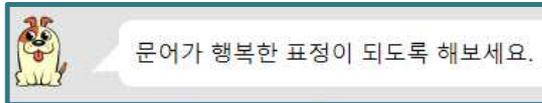
생활주제	동식물과 자연	활동형태	대·소집단활동-컴퓨터영역
활동목표	<ul style="list-style-type: none"> <li>· 해결하고자하는 문제를 정확히 인식하고 해결할 수 있다.</li> <li>· 컴퓨터를 이용한 간단한 코딩프로그램을 경험해본다.</li> <li>· 문어의 생김새와 특징을 안다.</li> </ul>		
누리과정 관련요소	<ul style="list-style-type: none"> <li>· 신체운동·건강 &gt; 신체 조절과 기본 운동하기 &gt; 기본 운동하기</li> <li>· 의사소통 &gt; 듣기 &gt; 바른 태도로 듣기</li> <li>· 자연탐구 &gt; 수학적 탐구하기 &gt; 규칙성 이해하기</li> </ul>		
소프트웨어 교육과정 관련요소	<ul style="list-style-type: none"> <li>· 알고리즘 라이프 &gt; 추상화 &gt; 해결하고자하는 문제를 인식하기</li> <li>· 알고리즘 라이프 &gt; 알고리즘 &gt; 문제 해결의 절차를 간단한 그림이나 기호로 표현하기</li> <li>· 알고리즘 라이프 &gt; 프로그래밍 &gt; 프로그래밍 도구 경험하기</li> </ul>		
활동자료	보석스티커, 화살표 모양의 스티커, 활동지		

#### 활동방법

문제발견이해	자료수집	자료분석	자료표현	문제분해	추상화	알고리즘	자동화	시뮬레이션
--------	------	------	------	------	-----	------	-----	-------

**1. 컴퓨터 화면에서 소개하고 있는 문어를 살펴보고, 활동의 목적을 알아본다.** 추상화

- 문어가 주인공인 컴퓨터 프로그램을 살펴보자.
- 여기 적혀 있는 문장을 같이 읽어볼까?



- 어떤 문제를 해결하는 것이 목적인지?
- 어떻게 하면 문어가 행복한 표정이 될까?

**2. 컴퓨터 화면에 보석스티커를 붙이고, 문어가 찾을 수 있도록 길을 만들어본다.**

**알고리즘**

- 여기에서는 우리가 길을 아무렇게나 만들어도 문어가 행복한 표정이 되는구나.
- 우리는 보석을 찾아야 행복해진다는 약속을 만들어보자.

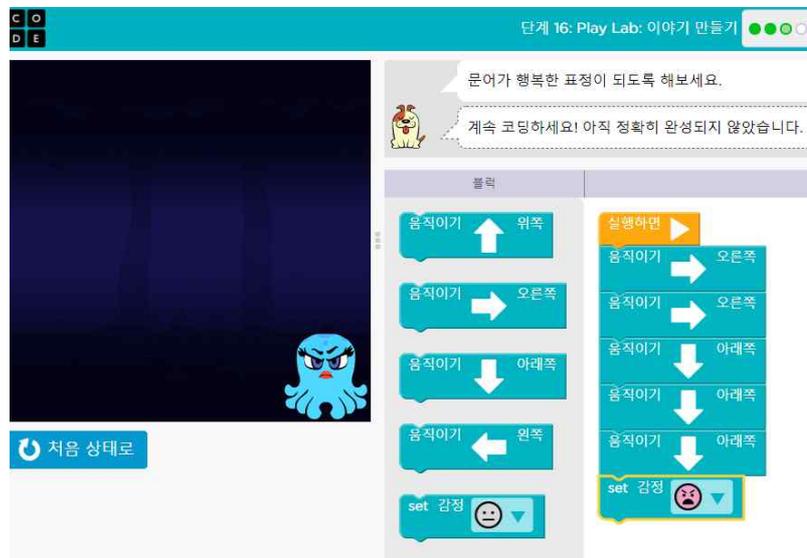
- 컴퓨터 화면 어디에 보석스티커를 붙여볼까? 누가 나와서 붙여볼래?
- 보석이 있는 곳까지 가려면 어떻게 움직여야 할까?
- 너희들이 말하면, 선생님이 먼저 시범을 보여줄게.
- 보석이 있는 곳까지 더 빨리 가려면 어떻게 해야 할까?
- 보석이 있는 곳까지 더 멀리 돌아가려면 어떻게 해야 할까?

**3. 화면의 블록들을 놓아보고, 문어가 보석을 잘 찾아가는지 실행버튼을 눌러 확인한다.**

자동화

시뮬레이션

- 우리가 길을 잘 만들었는지 실행 버튼을 눌러 확인해보자.
- 보석을 잘 찾아갔니?



- 어떤 부분이 잘못되어서 찾아가지 못했을까?
- 잘 찾아갈 수 있도록 다시 한 번 길을 만들어보자.



**[활동1] 소그룹 활동**

**4. 다른 길을 만들어 활동 이어가기**

- 친구들과 또 다른 길을 만들어 문어가 찾아갈 수 있는 방법을 생각해보고 블록을 쌓아보자.

**[활동2] 소그룹 활동**

**5. 단계별로 활동 이어가기**

- 꿀벌이 꽃을 모아서 꿀을 만들 수 있게 길을 찾아가게 도와주자.
- 1단계부터 할 수 있는 단계만큼 친구들과 서로 번갈아가면서 해보자.
- 어려워하는 친구가 있으면 옆에서 도와주자.

**6. 컴퓨터 화면의 블록들로 길을 만들면 편리한 점을 생각해보고, 더 해보기로 한다.**

**시뮬레이션**

- 문어가 보석을 찾아 행복한 표정이 될 때 어떤 기분이 들었니?
  - 꿀벌이 꽃을 모으고 꿀을 만들 때 어떤 기분이 들었니?
  - 이렇게 움직이는 길을 블록들로 표현하면 어떤 점이 좋을까?
- 예) 내가 만든 길을 다른 사람에게 쉽게 설명할 수 있다.  
여러 가지 길을 만들고 비교할 수 있다.

**7. code.org 사이트를 안내하고 가정에서 부모님과 함께 해 볼 수 있도록 한다.**

- 하원 후 집에서 부모님과 사이트에 접속해 다른 활동들을 함께 해 보고 다음시간에 경험에 대해 이야기 나누는 시간을 가지도록 하자.
- 단, 스마트 폰에서 하루에 1시간 이상 활동을 이어하지 않도록 주의사항을 안내해준다.

**참고사항**

- 컴퓨터 영역의 공간에 따라 소그룹으로 나누어 수업할 수 있다.
- 모든 유아들이 골고루 경험해볼 수 있도록 순서와 시간을 정해서 활동한다.

**가정과의 연계**

**소프트웨어 교육을 도와주는 웹사이트**

- \* 소프트웨어 교육에 관한 정보를 주고, 간단한 소프트웨어 프로그램을 경험할 수 있는 웹사이트를 소개합니다.
- ① '소프트웨어야 놀자' (<https://www.playsw.or.kr/main>)
  - 커넥트 재단이 무료로 운영하는 곳으로 가정에서 소프트웨어에 대한 이해를 돕는 동영상이나 놀이, 용어 설명 등이 보기 쉽게 제시되어 있어요.
- ② 'EBS 소프트웨어' (<http://home.ebs.co.kr/software/etc/8/htmlMenu>)
  - EBS에서 방영하는 각종 소프트웨어 교육 관련 자료들을 볼 수 있어요.

- ③ '코드닷오알지' (<https://code.org>)
  - 수준별 코딩 프로그램을 무료로 체험할 수 있어요.
- ④ '엔트리' (<https://playentry.org>)
  - 소프트웨어를 배울 수 있는 학습 사이트예요. 동영상 강의로 배우거나 간단한 프로그램 만들기를 할 수 있어요.
- ⑤ '스크래치' (<https://scratch.mit.edu>)
  - 이야기 만들거나 게임 프로그래밍 등 다양한 프로그램이 세계적으로 공유되는 곳이에요.

**<선생님을 위한 도움말>**

**♣. 초등컴퓨팅교사협회 (<http://hicomputing.org>)**

초등컴퓨팅교사협회는 소프트웨어 교육에 대해 연구하는 초등학교 선생님들이 만든 웹사이트이다. SW교육과정, 수업지도안, 언플러그드 교구와 교재, 피지컬&메이커 교육 등 각종 소프트웨어 교육에 관한 다양한 정보를 얻을 수 있다.



## 28. 생활 속에서 함께 하는 로봇들 알아보기 (로봇 프로그래밍하기 ①)

생활주제	생활도구	활동형태	대·소집단활동-이야기나누기
활동목표	<ul style="list-style-type: none"> <li>· 스택카드를 이용해 순차적으로 로봇에서 명령을 내릴 수 있다.</li> <li>· 순차적으로 명령을 내리지 않을 경우 로봇이 명령을 알지 못하는 것을 알 수 있다.</li> </ul>		
누리과정 관련요소	<ul style="list-style-type: none"> <li>· 자연탐구 &gt; 탐구하는 태도 기르기 &gt; 탐구과정 즐기기</li> <li>· 자연탐구 &gt; 과학적 탐구하기 &gt; 간단한 도구와 기계 활용하기</li> </ul>		
소프트웨어 교육과정 관련요소	<ul style="list-style-type: none"> <li>· 디지털컬처 &gt; 정보기기와 소프트웨어 &gt; 나에게 필요한 소프트웨어 선택하고 활용하기</li> <li>· 알고리즘라이프 &gt; 자료 &gt; 필요한 자료 찾아보기</li> <li>· 플레이컴퓨팅 &gt; 피지컬 컴퓨팅 &gt; 로봇에 적용할 프로그램 만들기</li> </ul>		
활동자료	알버트, 명령카드, 로봇관련 영상, 활동지		

### 활동방법

문제발견이해	자료수집	자료분석	자료표현	문제분해	추상화	알고리즘	자동화	시뮬레이션
--------	------	------	------	------	-----	------	-----	-------

#### 1. 로봇을 본 적이 있는지 경험에 대해 이야기 나눈다. 문제발견이해

- 로봇을 본 경험이 있니? 있다면 친구들에게 이야기 해 볼까?
- 로봇은 어떻게 움직이고 우리가 하는 말을 어떻게 알아들을까?
- 로봇과 이야기 해 보고 싶은 생각이 드니?
- 우리 주변에는 어떤 로봇들이 우리를 도와주고 있을까?

#### 2. 로봇이 이해할 수 있도록 명령을 내릴 수 있는 방법에 대해 이야기 나눈다.

자료수집 자료분석

- 로봇에게 명령을 내리려면 어떻게 해야 할까?
- 로봇에 관련된 짧은 영상을 보여주고 생각나는 이야기가 있는지 물어보고 같이 이야기를 나눈다.
- 이렇게 명령하고 싶은 카드를 순서대로 놓아 입력하는 것을 '코딩'이라고도 부른다

다.

### [활동1] 소그룹 활동

3. 알람을 사용해 본 적이 있는지 이야기 해보고 로봇을 알람시계로 만들 수 있는지 알아보자. 자료수집 자료분석

- 알람은 언제사용하고 어떻게 작동하는지에 대해 이야기 해보자.
- 로봇을 알람시계로 만들기 위해서는 어떤 명령을 내려야 하는지 머릿속에 떠올려 보자.

4. 로봇에게 명령하여 알람시계를 만들어 보기로 한다. 자료표현

- 로봇에게 어떤 것을 명령하고 싶니?  
예) 1분이 되면 색깔로 알려주는 로봇  
1분이 되면 소리로 알려주는 로봇
- 명령들의 순서를 이야기 해보고 카드에 그려보거나 적어서 순서대로 놓아보자.
- 명령카드를 알버트에게 순서대로 입력하고 동작하는지 확인해보자.

### [활동2] 소그룹 활동

3. 신호등이 어떻게 작동하는지 이야기 해보고 로봇을 신호등으로 만들 수 있을지 알아보자. 자료수집 자료분석

- 길을 건널 때 신호등은 어떻게 작동하고 언제 길을 안전하게 건널 수 있는지에 대해 이야기 해보자.
- 로봇을 신호등으로 만들기 위해서는 어떤 명령을 내려야 하는지 머릿속에 떠올려 보자.
- 손을 들었을 때 길을 건너도 좋다는 소리가 나오도록 하려면 어떤 명령을 내려야 하는지 머릿속에 떠올려 보자.

4. 로봇에게 명령하여 신호등을 만들어 보기로 한다. 자료표현

- 명령들의 순서를 이야기 해보고 카드에 그려보거나 적어서 순서대로 놓아보자.
- 명령카드를 알버트에게 순서대로 입력하고 동작하는지 확인해보자.

<b>참고사항</b>	- 자유선택활동 시간에 자유롭게 로봇에게 명령카드를 입력해보는 경험을 충분히 하도록 내어준다.
<b>가정과의 연계</b>	<div style="background-color: #cccccc; padding: 5px; display: inline-block;">집에 있는 도구들을 비교해보아요.</div> * 일상생활에서 사용하는 도구들의 장·단점을 비교하다보면, 새로운 생각이 떠오르고 발명의 계기가 되기도 하지요. 소프트웨어교육에서도 가장 처음은 문제의 발견입니다.

\* 늘 사용했던 것이지만 물건들을 새롭게 탐색하며 자녀와 이야기 나눠 보세요.

- 자를 때 사용하는 도구끼리 (가위, 칼, 펀치 등)
- 붙일 때 사용하는 도구끼리 (물풀, 목공용풀, 본드 등)
- 청소도구끼리 (빗자루, 청소기, 로봇청소기, 막대걸레 등)

예) 청소도구

물건	장점	단점

⇒ 어떤 물건이 발명되면 좋을까?

**<선생님을 위한 도움말>**

**♣. 유아기, 로봇과의 첫 만남은 어떻게?**

유아부터 초·중등학교 학생들을 대상으로 한 다양한 코딩로봇들이 개발되고 있어요. 코딩로봇의 사용방법을 가르쳐주는 곳도 많지요. 그런데 로봇 자체가 학습의 목적이 될 수는 없어요. 즉, 우리는 로봇을 익히기 위해서 공부하고 사용하는 것이 아니라, 우리 생활을 더 편리하게 하는 방법을 생각하다 보니 로봇을 이용하고 싶어지고, 그래서 로봇을 다루는 법을 배우게 되는 것입니다.

학급에서 아이들과 로봇을 다룰 때에도 로봇에게 무엇을 명령하면 우리 생활이 더 즐겁고 편해질까, 어떤 기능이 더해진다면 좋을까 등의 궁극적 목적부터 인식하도록 지도하는 것이 중요합니다.

## 29. 로봇에게 무엇을 명령할까? (로봇 프로그래밍하기 ②)

생활주제	생활도구	활동형태	대·소집단활동-과학					
<b>활동목표</b>	<ul style="list-style-type: none"> <li>· 생활을 더 편리하게 하기 위해 로봇을 사용해보는 경험을 한다.</li> <li>· 여러 가지 명령카드 중 나에게 필요한 것을 선택하여 사용할 수 있다.</li> <li>· 로봇에게 원하는 일을 명령하기 위해 순서를 생각하며 코딩해본다.</li> </ul>							
<b>누리과정 관련요소</b>	<ul style="list-style-type: none"> <li>· 의사소통 &gt; 듣기 &gt; 바른 태도로 듣기</li> <li>· 사회관계 &gt; 다른 사람과 더불어 생활하기 &gt; 공동체에서 화목하게 지내기</li> <li>· 자연탐구 &gt; 과학적 탐구하기 &gt; 간단한 도구와 기계 활용하기</li> </ul>							
<b>소프트웨어 교육과정 관련요소</b>	<ul style="list-style-type: none"> <li>· 알고리즘라이프 &gt; 추상화 &gt; 해결하고자하는 문제를 인식하기</li> <li>· 플레이컴퓨팅 &gt; 피지컬 컴퓨팅 &gt; 로봇의 사용 방법 및 주의 사항 알기</li> <li>· 플레이컴퓨팅 &gt; 피지컬 컴퓨팅 &gt; 프로그램에 따른 로봇 동작 확인하기</li> </ul>							
<b>활동자료</b>	코딩로봇, 명령카드							
활동방법								
<b>문제발견이해</b>	<b>자료수집</b>	<b>자료분석</b>	<b>자료표현</b>	<b>문제분해</b>	<b>추상화</b>	<b>알고리즘</b>	<b>자동화</b>	<b>시뮬레이션</b>
<p><b>1. 로봇을 이용하기 위해 무엇을 명령할지 생각해본다.</b> <span style="float: right;">자료표현</span></p> <ul style="list-style-type: none"> <li>- 로봇에게 어떤 것을 명령하고 싶니?</li> <li>예) 그대로 멈춰라 놀이</li> </ul>				<p><b>2. 로봇에게 어떤 기능을 명령할 수 있는지 알아본다.</b> <span style="float: right;">문제분해</span></p> <ul style="list-style-type: none"> <li>- 어떤 것들을 명령해야 로봇이 멈추지 않고 계속해서 움직일 수 있을까?</li> </ul>				
<p><b>3. 우리가 원하는 기능 중 로봇에게 명령할 수 있는 기능을 찾아본다.</b> <span style="float: right;">문제분해</span></p> <ul style="list-style-type: none"> <li>- 이 중에서 우리에게 필요한 명령카드는 무엇일까?</li> <li>- 우리가 하고 명령하고 싶지만, 명령카드가 없어서 못하는 것도 있을까?</li> </ul>				<p><b>4. 로봇에게 명령카드로 코딩해본다.</b> <span style="float: right;">추상화    알고리즘</span></p> <ul style="list-style-type: none"> <li>- 로봇이 앞뒤로 움직이게 하려면 어떤 명령카드를 사용해야 할까?</li> <li>- 로봇에게 손을 가져가면 멈추게 하려면 어떤 명령카드를 사용해야 할까?</li> </ul>				

- 명령하고 싶은 순서대로 카드를 놓아 보자.

**5. 명령한 기능을 반복되게 하려면 어떻게 해야 할지 알아보고 명령한다.** 자동화

- 로봇이 앞뒤로 움직이는 걸 계속 반복하게 하려면 어떤 명령카드를 사용해야 할까?
- 계속 반복하는 걸 멈출 필요가 있을 때는 어떤 명령카드를 사용해야 할까?
- 입력된 결과와 다르게 나와도 당황하지 말고 다시 해보자.
- 다르게 나왔을 때 입력한 본인의 카드 순서를 함께 이야기 해보자

**[소그룹 활동]**

**6. 그대로 멈춰라! 놀이를 상상하며 코딩 해본다.** 알고리즘 자동화

- 머릿속으로 그대로 멈춰라! 놀이를 상상하며 카드를 순차적으로 놓고 로봇에게 입력했을 때 어떻게 움직일지 생각해보자.
- 로봇에게 순차적으로 카드를 입력하고 결과를 확인해보자.
- 직접 사용해보고, 어떤 점이 잘됐는지, 힘들었던 점은 무엇이었는지 찾아보기로 하자.
- 다른 놀이를 로봇과 함께 한다면 어떤 걸 하고 싶은지 이야기 해 보자.

<b>참고사항</b>	<ul style="list-style-type: none"> <li>- 본 활동을 하기 전에 미리 자유선택활동 시간에 로봇에게 명령카드를 입력해보는 경험을 충분히 할 수 있도록 한다.</li> <li>- 먼저 각 명령 카드의 기본 기능을 익히고, 반복하기 기능 등을 사용해보도록 순차적으로 접근한다.</li> <li>- 유아들과 함께 정한 코딩 순서는 사진으로 찍거나, 기록해두어 다음 시간에 이어서 수업할 때 사용한다.</li> </ul>
-------------	---

<b>가정과의 연계</b>	<div style="background-color: #cccccc; padding: 5px; margin-bottom: 10px;"><b>유아기, 로봇과의 첫 만남은 어떻게?</b></div> <ul style="list-style-type: none"> <li>* 유아부터 초·중등학교 학생들을 대상으로 한 다양한 코딩로봇들이 개발되고 있어요. 코딩로봇의 사용방법을 가르쳐주는 곳도 많지요. 그런데 로봇 자체가 학습의 목적이 될 수는 없어요. 즉, 우리는 로봇을 익히기 위해서 공부하고 사용하는 것이 아니라, 우리 생활을 더 편리하게 하는 방법을 생각하다 보니 로봇을 이용하고 싶어지고, 그래서 로봇을 다루는 법을 배우게 되는 것입니다.</li> <li>* 가정에서도 아이들이 로봇을 만날 기회(예: 로봇청소기로 청소하기)가 있다면, 로봇에게 무엇을 명령하면 우리 생활이 더 즐겁고 편해질까, 어떤 기능이 더해진다면 좋을까 등의 궁극적 목적부터 인식하도록 지도하는 것이 중요합니다.</li> </ul>
----------------	--

**<선생님을 위한 도움말>**

### ♣. 코드(Code)

코드는 컴퓨터가 알아들을 수 있는 언어로, 컴퓨터가 해야 할 일을 알려주는 명령들을 모아놓은 것을 말합니다. 광범위하게 보면 특정한 형태의 정보를 표현하는 규칙으로, 우리가 사용하는 언어도 코드의 일종이라고 볼 수 있지요. 또 바코드, QR코드 등도 우리에게 친숙한 코드의 종류입니다.

그리고 컴퓨터는 0과 1 두 가지 숫자를 조합한 이진 코드를 사용해요. 자료를 처리하기 위해 일정한 규칙에 따라 코드를 만드는 것이 코딩이며, 컴퓨터가 복잡한 일을 빠르게 처리할 수 있게 코딩하려면 먼저 알고리즘을 구체적으로 만드는 일이 중요해요.

### 30. 길을 찾아가는 걸 도와줘, 로봇! (로봇 프로그래밍하기 ③)

생활주제	생활도구	활동형태	대·소집단활동-이야기나누기
활동목표	<ul style="list-style-type: none"> <li>· 일상생활 속의 문제를 비판적으로 생각해보는 경험을 한다.</li> <li>· 코딩한 로봇을 사용해보고 잘못된 점을 찾아서 수정할 수 있다.</li> <li>· 우리 생활을 더욱 편리하게 할 수 있는 방법을 창의적으로 생각해본다.</li> </ul>		
누리과정 관련요소	<ul style="list-style-type: none"> <li>· 신체운동·건강 &gt; 건강하게 생활하기 &gt; 건강한 일상생활하기</li> <li>· 자연탐구 &gt; 탐구하는 태도 기르기 &gt; 호기심을 유지하고 확장하기</li> <li>· 자연탐구 &gt; 과학적 탐구하기 &gt; 간단한 도구와 기계 활용하기</li> </ul>		
소프트웨어 교육과정 관련요소	<ul style="list-style-type: none"> <li>· 알고리즘라이프 &gt; 프로그래밍 &gt; 프로그래밍 도구 경험하기</li> <li>· 알고리즘라이프 &gt; 프로그래밍 &gt; 순차구조, 반복구조, 선택구조 경험하기</li> <li>· 플레이컴퓨팅 &gt; 피지컬 컴퓨팅 &gt; 로봇 프로그램 결과물 평가하기</li> </ul>		
활동자료	코딩로봇, 명령카드, 길 찾기 지도		

#### 활동방법

문제발견이해	자료수집	자료분석	자료표현	문제분해	추상화	알고리즘	자동화	시뮬레이션
--------	------	------	------	------	-----	------	-----	-------

#### [언플러그드 활동]

1. 길 찾기 지도의 중간에 친구가 서 있을 거야. 유치원까지 가려면 어떤 명령을 내려야

할지 머릿속으로 생각해 보자. 문제발견이해

- 왼쪽, 오른쪽, 앞, 뒤 명령카드를 사용해 길을 찾아 갈 수 있을까?
- 지난 시간에 배운 반복 카드를 사용할 수 있을까?

2. 문제점에 대해 이야기 나누고, 수정할 부분은 다시 명령한다. 시뮬레이션

- 혹시 불편한 점이나 바뀌서 다시 명령하고 싶은 부분도 있었니?
- 왜 그렇게 생각했니?
- 다시 코딩하면 고칠 수 있는 부분도 있니?
- 지금 다시 코딩해보자.

#### [소그룹 활동]

3. 이번에는 알버트가 소방서, 병원 등으로 찾아 갈 수 있게 명령을 내리려면 어떻게 해

**야 할까?** 문제발견·이해

- 알버트가 있는 방향이 어느 쪽인지 먼저 확인을 해 보자.
- 알버트에게 명령을 내려 보자.
- 친구가 명령카드를 사용하는 방법이 내가 나열한 것과 어떻게 다른지 살펴보자.

**4. 우리 주변에서 자주 사용되는 또 다른 로봇들에 대해 이야기 나누다.** 문제발견·이해

- 로봇을 코딩해서 생활에 이용해보니 어땠니?
- 어떤 점이 더 편리해졌니? / 불편했니?
- 우리 주변에는 또 어떤 로봇들이 우리를 도와주고 있을까?

<b>참고사항</b>	- 유아들의 창의적인 생각은 화이트보드에 적거나 메모해두었다가 조형영역에서 그림이나 만들기로 표현해보게 할 수 있다.
-------------	---

<b>가정과의 연계</b>	<p><b>자녀의 아이디어를 공유해주세요.</b></p>
	<ul style="list-style-type: none"> <li>* 일상생활 속의 문제들에 대해서 자녀와 이야기 나누다보면, 아이들이 멋진 생각을 해낼 때가 있지요?             <ul style="list-style-type: none"> <li>- (생활 속에서) 어떤 점이 불편하니?</li> <li>- 어떻게 바꿔보고 싶니?</li> <li>- 어떤 기능을 가진 물건이 발명되면 좋을까?</li> </ul> </li> <li>* 아이들이 해낸 재미있고 기발한 생각들은 가족들 또는 친구들과 나누었을 때 더욱 빛이 나고, 또 아이디어가 더해질 수도 있어요.             <ul style="list-style-type: none"> <li>- ○○이가 이런 생각을 해냈어! 어떻게 생각하니?</li> </ul> </li> </ul>

**<선생님을 위한 도움말>**

**♣. 피지컬 컴퓨팅(Physical Computing) 교구**

피지컬 컴퓨팅은 디지털 장치가 정보를 입력받고 실제 세계와 연결되어 결과를 출력하는 것을 의미합니다. 피지컬 컴퓨팅은 컴퓨터와 프로그램이 작동하는 방식을 배우기에 편리하며 간소화된 코딩 프로그램을 이용한 수 있는 다양한 피지컬 컴퓨팅 교구들이 개발되었습니다. 피지컬 컴퓨팅 교육을 위한 교구 몇 가지의 예를 보면 다음과 같다.

구분 교구	특징	제조사 or 수입처	공식홈페이지
<p>UO알버트</p> 	<ul style="list-style-type: none"> <li>· 스택카드 코딩</li> <li>· 엔트리/스크래치 연동</li> <li>· 전용 어플리케이션, 근접센서</li> <li>· LED눈으로 감정표현</li> </ul>	<p>SK telecom</p>	<p><a href="http://www.albert.school/main/index">http://www.albert.school/main/index</a></p>

구분 교구	특징	제조사 or 수입처	공식홈페이지
비봇 	<ul style="list-style-type: none"> <li>· 영국 90%이상의 유치원에서 사용하는 학습용 교구</li> <li>· 언플러그드 활동 제공</li> </ul>	에코스쿨스	<a href="https://www.beebot.org">https://www.beebot.org</a>
대시앤닷 	<ul style="list-style-type: none"> <li>· 미국 초등학교에서 주로 사용</li> <li>· 다양한 앱 제공(원더, 블록키)</li> <li>· 라이트, 사운드, 마이크(박수소리와 목소리 인식), 거리센서</li> </ul>	(주)인포마크	<a href="http://www.dashanddot.kr">http://www.dashanddot.kr</a>
오조봇 	<ul style="list-style-type: none"> <li>· 색을 인식하여 색 조합으로 명령</li> <li>· 마커펜과 종이로 다양한 컬러코드를 이용하여 명령 가능</li> <li>· 전용 앱 (Evo by ozobot)</li> <li>· 오조 블록클리 호환</li> </ul>	Marusys	<a href="http://www.ozobot.co.kr">http://www.ozobot.co.kr</a>
뚜루뚜루 	<ul style="list-style-type: none"> <li>· 카드, 앱, 블록 코딩 연동</li> <li>· 모션 인식</li> <li>· 라인트레이싱(그려진 선을 인식)</li> <li>· 홈페이지에서 액티비티 북 다운로드 가능</li> <li>· 뚜루봇 스크래치와 엔트리까지 활용</li> </ul>	시공미디어	<a href="http://www.truebot.com">http://www.truebot.com</a>