



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](#)

2023년 2월
박사학위 논문

언플러그드 교육과 EPL 교육이 지적장애
초등학생의 문제해결력, 수업 참여행동 및
컴퓨팅 사고력에 미치는 상대적 효과 분석

조선대학교 대학원

특수교육학과

채 성 은

언플러그드 교육과 EPL 교육이 지적장애 초등학생의 문제해결력, 수업 참여행동 및 컴퓨팅 사고력에 미치는 상대적 효과 분석

Analyses of the Relative Effects of Unplugged Education and EPL Education
on the Problem-solving Ability, Class Participatory Behaviors, and Computational
Thinking of Elementary School Students with Intellectual Disabilities

2023년 2월 24일

조선대학교 대학원

특수교육학과

채 성 은

언플러그드 교육과 EPL 교육이 지적장애
초등학생의 문제해결력, 수업 참여행동 및
컴퓨팅 사고력에 미치는 상대적 효과 분석

지도교수 허유성

이 논문을 교육학 박사학위신청 논문으로 제출함

2022년 10월

조선대학교 대학원

특수교육학과

채 성 은

채성은의 박사학위논문을 인준함

위원장 조선대학교 교수 김정연 (인)

위원 조선대학교 교수 김영일 (인)

위원 광주교육대학교 교수 허승준 (인)

위원 남부대학교 교수 김태강 (인)

위원 조선대학교 교수 허유성 (인)

2023년 1월

조선대학교 대학원

목차

ABSTRACT	vii
I . 서론	1
1. 연구의 필요성 및 목적	1
2. 연구 문제	8
3. 용어 정의	8
II . 이론적 배경	11
1. 장애학생과 SW교육	11
가. SW교육 개념 및 동향	11
나. 장애학생의 SW교육	12
2. SW교육 방법	14
가. 언플러그드 교육	15
나. EPL 교육	19
다. 언플러그드 교육과 EPL 교육의 차이	24
3. 장애학생 대상 SW교육의 효과	28
가. 문제해결력	28
나. 수업 참여	31
다. 컴퓨팅 사고력	37
4. 장애학생 대상 SW교육 관련 선행연구 분석	39
III . 연구 방법	46
1. 연구 설계	46
2. 연구참여자	47
3. 독립변인	51

가. 중재 절차	51
나. 연구 기간	60
다. 실험 장소	61
라. 중재 충실도	61
마. 사회적 타당도	63
4. 종속변인	64
가. 문제해결력	64
나. 수업 참여행동	69
다. 컴퓨팅 사고력	70
라. 관찰자 간 신뢰도	74
IV. 연구 결과	76
1. 언플러그드 교육과 EPL 교육이 문제해결력에 미치는 효과 비교	76
가. 도전 횟수 비교	77
나. 정답 도달 시간 비교	79
다. 1회 평균 구상 시간 비교	82
라. 오류점검 횟수 비교	85
마. 효율적인 알고리즘 구상 비교	85
2. 언플러그드 교육과 EPL 교육이 수업 참여행동에 미치는 효과 비교	88
가. 자발적 질문 비교	89
나. 주의집중 행동 비교	91
다. 과제회피 행동 비교	93
3. SW교육이 컴퓨팅 사고력에 미치는 효과	96
V. 논의 및 제언	100
1. 논의	100
가. 언플러그드 교육과 EPL 교육이 문제해결력에 미치는 효과 비교에 대한 논의	101
나. 언플러그드 교육과 EPL 교육이 수업 참여행동에 미치는 효과 비교에 대한 논의	105

다. SW교육이 컴퓨팅 사고력에 미치는 효과에 대한 논의	106
2. 연구의 제한점	108
3. 제언	109
참고문헌	111
부 록	130

표목차

<표 II-1> ‘알고리즘과 프로그래밍’ 영역의 성취 기준	22
<표 II-2> 두 교육 방법의 차이	27
<표 II-3> 선행연구의 수업유형, 종속변인, 주요 결과	43
<표 III-1> 종속변인별 실험설계 및 분석방법	47
<표 III-2> 연구참여자 기본 정보 및 특성	48
<표 III-3> 언플러그드 교육과 EPL 교육의 회기별 교육 활동	52
<표 III-4> 기초선 기간 교육내용	53
<표 III-5> 두 교육 방법의 차이	54
<표 III-6> 언플러그드 교육 활동내용	55
<표 III-7> 언플러그드 교육 예시 지도안	56
<표 III-8> EPL 교육 활동내용	57
<표 III-9> EPL 교육 예시 지도안	59
<표 III-10> 효과적인 처치 구간의 교육 내용	60
<표 III-11> 중재 충실도 측정 항목	62
<표 III-12> 수업별 중재 충실도의 평균과 범위	62
<표 III-13> 사회적 타당도 평균과 범위	63
<표 III-14> 문제해결력 조작적 정의	66
<표 III-15> 수업 참여행동 조작적 정의	70
<표 III-16> 컴퓨팅 사고력 세부 요소	72
<표 III-17> 비버챌린지 문항별 컴퓨팅 사고력 세부 구성 요소	74
<표 III-18> 관찰자 간 신뢰도의 평균과 범위	75
<표 IV-1> 전체 학생의 중재별 평균 문제해결력 결과	76
<표 IV-2> 수업유형별 도전 횟수 및 평균	77
<표 IV-3> 수업유형별 정답 도달 시간 및 평균	80
<표 IV-4> 수업유형별 1회 평균 구상 시간 및 평균	83
<표 IV-5> 수업유형별 오류점검 횟수	85
<표 IV-6> 수업유형별 효율적인 알고리즘 구상률 및 평균	86

<표 IV-7> 전체 학생의 구간별 평균 수업 참여행동 수치와 범위89
<표 IV-8> 전체 학생의 문제별 컴퓨팅 사고력 사전-사후 검사 점수96
<표 IV-9> 전체 학생의 컴퓨팅 사고력 요소 사전-사후 검사 점수97

그림목차

[그림 II-1] 엔트리 화면 구성	20
[그림 III-1] 알고리즘 문제 예시	68
[그림 III-2] 명령어 카드 및 명령어 블록	69
[그림 IV-1] 수업유형별 도전 횟수 그래프와 추세선	78
[그림 IV-2] 정답 도달 시간 그래프와 추세선	81
[그림 IV-3] 1회 평균 구상 시간 그래프와 추세선	84
[그림 IV-4] 효율적인 알고리즘 구상률 그래프와 추세선	87
[그림 IV-5] 자발적 질문 그래프와 추세선	90
[그림 IV-6] 주의집중 행동 그래프와 추세선	92
[그림 IV-7] 과제회피 행동 그래프와 추세선	95
[그림 IV-8] 학생별 컴퓨팅 사고력 사전-사후 검사 점수	97
[그림 IV-9] 학생별 컴퓨팅 사고력 하위요소 비교	98

ABSTRACT

Analyses of the Relative Effects of Unplugged Education and EPL Education on the Problem-solving Ability, Class Participatory Behaviors, and Computational Thinking of Elementary School Students with Intellectual Disabilities

Seong Eun Chae

Advisor: Prof. Yu-sung Heo, Ph.D.

Department of Special Education

Graduate School of Chosun University

Software education was included in the regular curriculum due to the 4th industrial revolution. Software education develops computational thinking, logical thinking ability, procedural thinking ability, reflective thinking ability, and problem solving ability. Therefore, education should be provided to both students with and without disabilities. It should not be limited to education for specific students only such as students without disabilities. Students with intellectual disabilities show difficulty in problem-solving due to limitations in their cognitive abilities. They also show difficulties in class participatory behaviors due to low motivation to learn and learned helplessness. Unplugged education can learn the concepts and principles of computer science through fun operation activities. EPL education shows algorithms designed on a computer screen. Therefore, software education can be used as a learning method to increase the interest and motivation of learners. Despite these advantages, the number of software education studies for students with disabilities is very small.

Therefore, in this study, unplugged education and EPL education were conducted for elementary school students with intellectual disabilities to analyses the relative effects seen on problem-solving ability and class

participatory behavior, and to find out the effect on computational thinking. The intervention was implemented using a alternating treatments design for three students with intellectual disabilities enrolled in a special class in an elementary school and the effect was confirmed.

The research results of this study are as follows.

First, EPL education was found to be more effective than unplugged education in the problem-solving ability and class participatory behavior of elementary school students with intellectual disabilities.

Second, for problem-solving ability, the number of challenges, the time to reach the correct answer, the average conception time per session, the number of error checks, and efficient algorithm design were measured. The results are as follows. Unplugged education was found to be effective in the number of challenges and efficient algorithm design. EPL education was found to be effective in the time to reach the correct answer and the average conception time per session. As for the number of error checks, the occurrence rate was low, so a meaningful comparison could not be made.

Third, in class participatory behavior, spontaneous questioning, attention-focused behavior, and task avoidance behavior were measured, and the results are as follows. EPL education was effective for attention-focused behavior and task avoidance behavior, and the effectiveness was proven through an effective treatment section. As for the spontaneous questioning, the occurrence rate was low, so a meaningful comparison could not be made.

Fourth, through SW education(unplugged education and EPL education), computational thinking was improved in two students. There was no difference in the pre-post results of computational thinking for one student. Among the sub-domains of computational thinking, in the algorithm element, all three students showed an increase in post results.

At the discussion section, educational implications for special education teacher base on the results of the this study were discussed. In addition, the limitations of this study and suggestions for future study was presented.

1. 서론

1. 연구의 필요성 및 목적

과학 기술 발전의 영향은 현대 시민의 삶에 다양한 변화와 도전을 일으키고 있다. 알려진 혹은 알려지지 않은 기술 기반의 미래 사회에는 새로운 역량들이 필요할 것이며, 이러한 역량들은 가까운 미래에 개인과 사회 및 직업에까지 영향을 미치게 될 것이다. 특히 디지털로 대표되는 현재와 미래 사회는 소프트웨어로 구동되는 기술과 기기로 가득 차고, 인간들은 점점 더 컴퓨터 혹은 소프트웨어 기반의 세계에 묻혀 살게 될 것이다. 이러한 현실을 고려할 때 우리를 둘러싼 디지털 생태계를 이해하고 효과적으로 참여하기 위해 컴퓨터 언어를 이해하고 다루는 것은 필수 불가결한 일이 되었다(da Silva et al., 2020; Román-González et al., 2016).

교육은 새로운 시대적, 사회적 요구와 다음 세대 학생들이 통합되는 연결고리의 역할을 한다. 이 과정에서 컴퓨팅 사고력(computational thinking)은 다음 세대를 위한 기본 역량 중 하나가 되고 있다(da Silva et al., 2020). 컴퓨팅 사고력이란 컴퓨터 과학의 원리를 활용해 문제를 효율적으로 해결하는 절차적 사고 능력(이신현, 2018)을 말하는데, 최근 주요 국가들은 컴퓨팅 사고력을 중요한 교육의 일환으로 도입하기 위해 국가 커리큘럼을 수정하고 있다(성정숙, 김현철, 2015; 이재은, 김준수, 2020; Brackmann et al., 2017). 유치원에서 고등학교까지 모든 수준에서 더 많은 컴퓨터 과학(computer science) 교육을 반영하여 운영하고 있다. 국가 수준의 의무 교육에 컴퓨팅 사고력을 통합시키는 이유는 간단하다. 첫째, 미래 고용을 준비하기 위해서다. 향후 미래 사회에서는 인공지능(AI), 메타버스 등 다양한 소프트웨어 기반, 기술 기반의 직업이 국가의 부출 창출하는 핵심 요인이 될 것이다. 둘째, 학생들이 다른 방식으로 생각하고 새로운 미디어를 사용하여 자신을 표현하고 실제 문제를 해결할 수 있는 역량을 개발하기 위한 경험을 제공하기 위해서다(Brackmann et al., 2017; Fagerlund et al., 2021).

제 4차 산업혁명 시대에 발맞춰 우리나라 역시 컴퓨팅 사고력을 가진 창의·융합 인재를 기르기 위해 2015 개정 교육과정에 소프트웨어 교육(이하 SW교육)

을 도입하였으며, 2017년부터 SW교육이 이루어지고 있다. SW교육이란 컴퓨터 과학의 기본적인 개념과 원리를 바탕으로 다양한 문제를 창의적이고 효율적으로 해결하는 컴퓨팅 사고력을 기르는 교육으로 알고리즘과 프로그래밍이 포함된 교육을 말한다(교육부, 2017a). 우리나라의 SW교육은 프로그램 개발 역량 자체에 초점을 맞추기보다는 정보윤리 의식과 태도를 바탕으로 실생활에서 접하는 문제를 컴퓨팅 사고로 해결할 수 있도록 교육하는 데에 역점을 두고 있다(교육부, 2015a). 특히 2022년 12월 22일 교육부(2022a)는 2022 개정 교육과정을 발표하였는데, 그 안에 인공지능(AI)과 SW교육을 강화하는 내용을 포함하였다. 초·중·고등학교 모두에서 인공지능과 관련된 내용을 신설 및 강화하였다. SW교육은 초등학교의 경우 「실과」 교과를 포함해 자율시간을 활용하여 34시간 이상, 중학교의 경우 「정보」 교과는 학교 자율시간을 확보해 68시간 이상 편성·운영할 것을 권장하였다. 고등학교의 경우에는 「정보」 교과를 신설하여 진로 적성에 따른 다양한 선택과목을 편성해야 한다. 2022 개정 교육과정에서는 SW교육의 시수가 2015 개정 교육과정보다 증가하였고 고등학교에 「정보」 교과가 신설되는 등 디지털 기초 소양 및 컴퓨팅 사고력 함양을 위해 「정보」 교육과정이 재구조화 되었다(교육부, 2022a; 교육부, 2021).

이와 더불어 기본 교육과정을 통해 장애학생의 SW교육에 대한 내용도 개정되었다. 초등학교 「실과」에서는 정보통신기기를 활용하여 정보를 사용하고 실생활의 문제를 절차적 사고로 해결할 수 있는 능력을 갖추는 데 중점을 두어 개정되었으며, 중·고등학교 선택교과인 「정보통신활용」에서는 컴퓨팅 사고력 증진에 중점을 두고, 알고리즘 접근을 통해 생활 속 문제를 해결하는 데 중점을 두어 개정되었다(교육부, 2022b). SW교육이 전문 엔지니어를 양성하는 목적보다는 누구나 기본적으로 컴퓨팅 사고력, 논리적 사고력, 문제해결력을 키울 수 있는 보편교육(전수진, 2017; 최유현, 2019; Wing, 2006)으로 인식되는 사회적 흐름이 반영된 결과로 보인다.

개정 교육과정을 통해 장애학생도 장애 정도와 학습 수준에 적합한 SW교육을 학습하여 다양한 삶의 장면에서 필수적인 컴퓨팅 사고력과 문제해결력 등을 함양할 수 있을 것이다. 장애학생 역시 컴퓨팅 사고력이나 문제해결력 등의 습득 수준에 상관없이 SW교육에서 경험할 수 있는 것을 경험해야 한다. 이는 향후 장애학생이 사회의 구성원으로 살아가는 데 유용하고 필요한 중요한 역량이 될 것이다. 따라서 장애학생을 대상으로 기존의 비장애학생 대상 SW교육을 어떻게

수정하여 적합화할 것인지, 어떤 SW교육이 장애학생에 더 접근하기 쉽고 효과적인지 등에 관한 체계적인 연구가 필요하다.

학습과 프로그래밍 과정은 일견 유사하다. 일반적으로 학습에서는 먼저 문제를 이해하고 분석한다. 그 후 문제를 풀어보고, 학습한 내용에 대해 무엇을 알고 무엇을 모르는지 평가한다. 부족한 부분을 재학습하고 오개념을 수정한다. 프로그래밍도 유사하다. 임희석(2017)은 학습과 프로그래밍의 유사점에 대해 다음 여섯 가지를 제시한다. 프로그래밍 과정은 첫째, 문제를 이해하고 분석한다. 둘째, 문제의 핵심 요소만 추출하여 추상화하고 추출된 요소로 문제를 재정의한다. 셋째, 문제해결을 위한 방법과 절차인 알고리즘을 고안한다. 넷째, 고안된 알고리즘대로 컴퓨터가 일을 수행할 수 있도록 프로그램을 작성하고 결과가 정확한지 확인한다. 다섯째, 만약 결과가 올바르지 않으면 무엇이 잘못되었는지 확인하고 오류를 수정한다. 여섯째, 목적이 완성될 때까지 위 단계를 반복한다. 따라서 임희석(2017)의 주장처럼 프로그래밍 경험을 통해 장애학생의 문제해결 등을 포함하는 고차적 사고력 능력의 개발이 가능하다.

지적장애 또는 자폐성장애 학생들은 학습에서 다양한 어려움을 경험한다. 특히 절차적 사고, 반성적 사고, 자기통제 등 고차적 사고력이 제한적이다(이옥인, 백은희, 2005). 일반적으로 고차적 사고력이란 특정 과제를 해결할 때 어떤 전략을 사용해야 하는지를 알고, 동기를 조절하고, 자신이 했던 일들을 검토하며, 결과의 효율성을 점검하고 노력의 결과를 점검하는 사고 능력을 말한다(송준만 외, 2012; 이소현, 박은혜, 2011). SW교육은 장애학생의 고차적 사고력의 계발에 긍정적인 영향을 미칠 수 있다. SW교육은 일반적으로 컴퓨팅 사고력, 논리적 사고력, 절차적 사고, 반성적 사고, 문제해결력 등 고등능력을 길러주는 교육 방법으로 알려져 있다(성영훈, 유승한, 2016; 윤선희, 2018; 정인재, 정석주, 2021; Bouck & Yadav, 2022). SW교육에서 ‘컴퓨팅 사고력’이란 단순히 컴퓨터가 문제를 해결하는 것과 같은 방식으로 문제를 해결해야 한다는 것이 아니라, 복잡한 문제를 효율적으로 다루고 해결하는 사고 능력이다. 즉, 이것은 컴퓨터가 문제를 해결하는 방식처럼 복잡한 문제를 단순화하고 이를 논리적, 효율적으로 해결하는 능력을 말한다(Voskoglou & Buckley, 2012). 따라서 컴퓨팅 사고력을 갖춘 장애학생은 실제 상황에서 복잡한 문제에 직면할 때, 문제를 여러모로 이해하고 보다 절차적이고 합리적인 선택을 하며 해결할 수 있을 것이다. SW교육은 다수의 연구를 통해 컴퓨팅 사고력(김성준, 2019;

윤선희, 2018; 이준형 외, 2018; 이철현, 2017; 조은진, 2022; González-González et al., 2019)뿐 아니라 논리적 사고력(이좌택, 이상봉, 2004; 정인재, 전석주, 2021)에도 효과가 있는 것으로 나타나 장애학생이 어려움을 겪는 고차원적 사고 향상에 도움을 줄 것으로 기대된다.

SW교육은 또한 장애학생의 문제해결력에 긍정적인 영향을 미칠 수 있다(강은희, 2019; 오지훈 외, 2021). 일반적으로 ‘문제’는 해결의 절차가 이미 알려져 단순한 문제보다는, 구체적이고 확실한 해결의 방법을 쉽게 구상하기 어렵고 해결하기 위해서는 여러 단계를 거친 다양한 사고가 요구되는 것을 의미한다(Polya, 1962). 이러한 문제는 우리의 일상에 다양하게 편재되어, 비장애 학생은 물론 장애학생 역시 수많은 문제 상황에 놓일 수밖에 없다. 문제해결력은 삶에서 매우 중요한 역량 중 하나이다. 더욱이 날로 증가하는 기술의 발전과 사회가 복잡하게 분화되면서 기술과 지식을 결합하여 문제를 해결하는 능력은 더욱 중요해지고 있다. 문제해결 능력을 갖춘 장애학생들은 주변의 다양한 문제에 대해 감정적이고 즉흥적으로 대응하기보다 논리적이고 절차적으로 접근하여 학습의 효율성을 높일 뿐 아니라 또래와 유기적이고 건강한 관계를 구축하게 된다(Hufad et al., 2021; Voskoglou & Buckley, 2012).

장애학생은 제한된 활동과 경험, 인지 능력의 부족, 학습 내용의 일반화와 전이의 어려움 등으로 인해 문제해결력에 어려움을 보이는 것으로 알려져 있다(박신영, 손승현, 2010; 이소현, 박은혜, 2011; APA, 2018). 장애학생의 문제해결력 향상을 위해 다양한 선행연구가 수행되었다. 주요 선행연구를 보면, 장애학생의 문제해결력 향상을 위한 중재로 전략교수와 관련된 접근이 주를 이뤘다(강옥려, 고승희, 2005; 구슬이, 신진숙, 2019; 김순영, 2002; 이필애, 2007; 이현정, 2022). 반면, 장애학생을 대상으로 SW교육을 활용한 문제해결력 연구는 매우 제한적이다. 비장애학생을 대상으로 실시된 선행연구(강은희, 2019; 권정인, 2014; 노지예, 이정민, 2018; 오지훈 외, 2021; 윤선희, 2018)에 따르면 SW교육은 문제해결력 또는 창의적 문제해결력의 향상에 도움이 되는 것으로 나타났으며, 문제 명료화, 최적인 선정, 계획 수립 및 실행 등의 문제해결력 영역에 효과가 있는 것으로 나타났다. SW교육을 장애학생에게 적용한다면 문제해결력 향상에 긍정적인 영향을 미칠 수 있을 것으로 기대된다.

지적장애 학생들은 낮은 지적 기능과 적응행동 기능의 제한(Robert et al., 2022)으로 인해 학교 수업 참여에 어려움을 보이는 경우가 많다(정지영, 202

2). 수업 참여행동이란 수업에서 이루어지는 과정에서 관찰되는 행동(변관석, 2016)을 말하며 적극적인 수업 참여는 교육의 성공과 학생의 개인 개발에 중요한 역할을 한다(Mustapha et al., 2010). Fassinger(1996)는 수업 참여를 열정적으로 수업에 참여하며 자유롭게 의견을 제시하거나 질문을 제기하는 행동으로 보았으며, Bippus & Young(2000)은 수업 참여를 학급 토론에 참여하고 부정적인 행동을 삼가는 것으로 정의하였다. 수업에 적극적으로 참여하는 것은 학습 성과를 향상시키고 학생의 학업 능력에 긍정적인 기대를 갖게 하는 지표(Bippus & Young, 2000; Furrer & Skinner, 2003)가 되므로 장애학생의 수업 참여행동에 초점을 맞추는 것은 중요하다. 선행연구를 살펴보면 장애학생들의 수업 참여행동을 높이기 위해 문제중심학습(PBL), 앵커드 수업, 신체활동 수업, 교수적 수정(변관석, 2016; 서민영, 정은희, 2021; 이경순, 2007; 이상철, 2012; 임해주, 2017) 등의 연구가 이루어졌으며 수업 참여행동에 긍정적인 결과를 나타냈다. 이처럼 장애학생의 개별적 요구를 반영한 흥미로운 수업을 통해 장애학생의 수업 참여의 질이 높아짐을 알 수 있다. SW교육은 언플러그드 교육을 통해 컴퓨팅 사고력을 높이 학습으로 재미있게 학습할 수 있으며 수업에 높은 몰입과 관심을 가지게 한다(교육부, 2017a). 또한 EPL 교육이나 피지컬 컴퓨팅 등의 교육은 컴퓨터, 태블릿 PC, 로봇, 드론 등을 이용하기 때문에 학생들의 흥미와 동기유발에 효과가 있는 것으로 나타나(교육부, 2017a; 이은경, 이영준, 2007) 장애학생의 적극적인 수업 참여를 강화할 수 있는 중재로써 사용될 수 있을 것으로 보인다.

SW교육의 교수·학습 방법은 다양하다. 언플러그드 활동(unplugged activities; 이하 언플러그드 교육)과 교육용 프로그래밍 언어(Educational Programming Language: EPL) 교육(이하 EPL 교육)은 학교 현장에서 가장 흔히 활용하는 SW교육 유형이다(강두봉, 2019; 문우종 외, 2020; 성영훈, 유승한, 2016; 신성희, 한경근, 2019). 언플러그드 교육이란 컴퓨터 과학의 기본 원리, 알고리즘적 사고, 프로그래밍 원리를 컴퓨터나 전문 장비를 사용하지 않고 게임이나 놀이를 통해 배우는 방법이다(김혜란, 최선영, 2019; 장정훈, 김종우, 2016; 전수진, 2017). Hufad 외(2021)와 Rodriguez 등(2017)은 언플러그드 교육이 컴퓨터나 타이핑 기술이 없는 학생들의 흥미를 끌 수 있는 유익한 활동이자 학생들의 초기 문제해결 능력을 길러줄 수 있는 좋은 전략이 된다고 하였다. 이는 놀이 형태의 체험 활동 및 구체물 등을 이용한 조작 활동으로 이루어지기 때문에

구체적 조작단계에 해당하는 장애학생들에게 특히 적합하다(한병래, 2013). 이러한 특성들로 인해 언플러그드 교육은 유아나 초등학교 저학년을 대상으로 한 연구(구영은, 2015; 김필현, 2021; 송한민, 2021; 이재호, 오상미, 2021)에서 긍정적인 효과가 보고되었다.

한편, EPL 교육은 학생들이 전통적인 프로그래밍 언어의 복잡한 구문을 배울 필요 없이 프로그래밍을 할 수 있게(Moreno-León et al., 2015) 교육용으로 제작한 프로그래밍 언어이다(서영호 외, 2016). 컴퓨팅 사고력을 가르치는 가장 일반적인 전략은 다양한 프로그래밍 작업을 기반으로 하는 컴퓨터 활동이며(Brackmann et al., 2017), EPL은 이러한 특성을 반영한다. EPL 교육은 알고리즘 사고 능력과 문제해결력을 신장시키며, 창조적으로 생각하고 다양한 방식으로 문제에 접근하며 체계적으로 문제를 해결하는 알고리즘 사고방식을 기르는 데 효과적인 방법으로 알려져 있다(조성환 외, 2008). 최근 EPL 교육은 컴퓨팅 사고력 증진을 위한 교육 방법으로 초·중·고등학교 전반에 걸쳐 교육이 이루어지고 있다. 초기에는 특정 프로그래밍 언어를 학습하는 데 교육의 목표를 두었으나 점차 문제해결 능력이나 창의적 사고를 향상하는 데로 교육목표가 바뀌고 있다(김시정, 조도은, 2018).

학교 현장에서 SW교육의 기초·기본 교수전략으로 활용되고 있는 언플러그드 교육과 EPL 교육의 특징을 간단히 비교하면 다음과 같다. 첫째, 언플러그드 교육은 말 그대로 플러그(컴퓨터) 없이 구체물과 놀이 및 조작 활동을 통해 학습이 이루어지므로(김혜란, 최선영, 2019; 최유현, 이승원, 2019; Bell et al., 2009) 어린 학습자, 즉 구체적 조작기에 해당하는 학생들에게 적합하다(한병래, 2013). 반면에 EPL 교육은 컴퓨팅 기기를 통해 학습이 이루어지며 학습의 결과가 애니메이션 효과로 즉각적이고 생동감 있게 나타나(이은경, 이영준, 2007; 이준형 외, 2018) 학생의 흥미와 관심을 유발할 수 있다. 나이가 어린 학습자들은 접근하기 어렵다는 지적도 있지만(나원영, 이철현, 2016), SW교육의 대중화로 유아를 대상으로 한 EPL 교육 연구도 이루어지고 있다(한지원, 2019; 황성신, 2019). 둘째, 언플러그드 교육의 활동내용은 교사가 인터넷 웹사이트(예: csunplugged.org)를 통해 교육자료를 쉽고 빠르게 무료로 접근할 수 있다. 반면, EPL 교육은 특정 EPL 교육 플랫폼(엔트리, 스크래치, Code.org 등)에 접속하여 이루어지는 교육으로 교사가 특정 플랫폼에 대한 기본 지식을 가지고 교수할 수 있어야 하므로(안창호 외, 2018) 별도의 교사 연수가 필요하

다. 셋째, 언플러그드 교육은 컴퓨팅 기기에 대한 학생의 선행지식이 필요 없지만(김혜란, 최선영, 2019; 최유현, 이승원, 2019; 한선관, 2010), EPL 교육은 컴퓨터의 마우스 조작법 혹은 태블릿 PC의 터치법, 드래그 앤 드롭 방식의 조작 방법 등의 미디어 문해력이 필요하다(전석주, 2021). 넷째, 언플러그드 교육은 주로 컴퓨터 과학의 원리나 개념 학습에 초점을 두고(권은정, 2008; 김혜란, 최선영, 2019), EPL 교육은 프로그래밍 교육(오지훈 외, 2021)에 초점을 둔다는 특징이 있다.

국가 차원의 SW교육에 대한 관심에도 불구하고 우리나라 학교 단위 SW교육은 아직 초기 단계에 있다. 기본 인프라를 갖춘 학교의 비율은 여전히 높지 않다(김한성, 전수진, 2019; 마대성, 2021). EPL 교육은 컴퓨터 기자재 및 네트워크 시설이 준비되어 있어야 한다는 단점이 있고(이철현, 2015), 언플러그드 교육 역시 교구의 구매뿐 아니라 전문 강사의 확보나 교사 연수 등 학교 차원의 관심과 예산 확보 및 시설 설비 확충이 되어 있지 않으면 충실한 교육이 이루어지기 어렵다.

일반교육 분야에서 SW교육의 연구 동향을 분석한 연구를 살펴보면, 연구의 목적 및 수집 범위에 따라 차이가 있지만, 황지은, 황성운(2017)의 연구에서는 72편, 김재경(2018)의 연구에서는 166편, 이해화(2018)의 연구에서는 총 340편을 분석하였다. 상당히 많은 수의 연구가 수행되었다는 것을 간접적으로 알 수 있다. 반면, 채성은, 허유성(2021)이 최근 10년간 국내외 특수교육 분야에서 진행된 SW교육 관련 연구 동향을 분석한 결과를 보면, 국내 논문은 총 17편으로 일반교육 분야와 비교해 관련 연구가 현저히 낮았다. 이 중 실험연구는 8편으로 매우 부족한 수준이다. 더욱이 코로나바이러스 감염증-19로 인해 학교 수업이 원격수업으로 전환되고 대면수업이 최소화되면서 놀이 중심의 언플러그드 교육 혹은 컴퓨팅 기기를 사용해야 하는 EPL 교육이나 피지컬 컴퓨팅 교육을 실시하기 어려워 최근에는 장애학생 대상 SW교육 관련 중재 연구가 주춤한 상황이다.

장애학생 대상 언플러그드 교육과 EPL 교육에 관한 연구를 살펴보면, 언플러그드 교육만을 활용한 중재 연구가 2편(김정일 외, 2012; 신성희, 한경근, 2019), EPL 교육만을 적용한 중재 연구가 1편(정소영, 2019), 그 외에 언플러그드 교육과 EPL 교육을 함께 적용한 연구가 3편(김소연, 민천식, 2019; 이현주 외, 2020; 홍다희, 2016)으로 관련 연구가 매우 적으며 두 중재 간에 장애학생

에게 미치는 상대적 교육 효과를 비교하거나 장애학생의 수업 참여 양상(예: 수업 참여행동 등) 등을 체계적으로 비교 분석한 연구는 찾아보기 힘들다. 특히 초등학교 지적장애 학생처럼 수업 중 주의집중력이 부족하고(곽은정, 신진숙, 2011; 송준만 외, 2012) 다양한 문제행동을 보이는 학생의 경우(곽은정, 신진숙, 2011; 문병훈 외, 2017), 문제해결력이나 컴퓨팅 사고력 등에 미치는 중재 간 효과를 분석하는 것도 중요하지만, 두 중재를 수행하는 동안 학생들이 얼마나 수업에 제대로 참여하는지(예: 주의집중 행동, 자발적 질문, 과제회피 행동 등)를 비교 분석하는 것도 교육적으로 중요한 의미를 가진다(황규용, 2017). 그러나 관련 연구는 매우 제한적이다.

이에 이 연구의 목적은 학교 현장에서 가장 활용도가 높고, 이미 교육적 효과가 검증된 SW교육 방법의 하나인 언플러그드 교육과 EPL 교육이 지적장애 초등학생의 문제해결력, 수업 참여행동에 미치는 상대적 효과를 분석하고 컴퓨팅 사고력에 미치는 영향을 분석하는 것이다. 언플러그드 교육과 EPL 교육은 각각 서로 다른 장단점을 가지고 있다. 이에 지적장애 초등학생의 특성에 따라 문제해결력이나 컴퓨팅 사고력뿐 아니라 수업 참여행동 등 서로 다른 영향이나 효과가 나타날 수 있다. 이 연구의 결과는 향후 초등학교에서 지적장애 학생을 대상으로 SW교육을 준비할 때, 장애학생의 특성에 더 적합한 효과적인 SW교육 방법(언플러그드 교육, EPL 교육)을 선택하는 데 유용한 정보를 제공할 수 있을 것이다.

2. 연구 문제

첫째, 언플러그드 교육와 EPL 교육이 지적장애 초등학생의 문제해결력에 미치는 상대적 효과는 어떠한가?

둘째, 언플러그드 교육과 EPL 교육이 지적장애 초등학생의 수업 참여행동에 미치는 상대적 효과는 어떠한가?

셋째, SW교육(언플러그드 교육과 EPL 교육)이 지적장애 초등학생의 컴퓨팅 사고력에 미치는 효과는 어떠한가?

3. 용어 정의

가. 언플러그드 교육

컴퓨터를 사용하지 않고 컴퓨터 과학의 원리와 개념을 배울 수 있는 교육 활동이며 컴퓨팅 사고력 신장을 위해 의도적으로 만들어진 교육 방법이다(교육부, 2017b). 이 연구에서 언플러그드 교육이란 컴퓨터나 기기를 사용하지 않고 구체적 조작 활동을 통해 배우는 컴퓨터 과학의 원리와 알고리즘의 기본 개념이 내재된 학습활동을 말한다.

나. EPL 교육

EPL은 교육용 프로그래밍 언어로서 교육을 목적으로 만들어진 학습용 프로그래밍 언어로(오지훈 외, 2021). 프로그래밍을 처음 접하는 학습자에게 블록처럼 생긴 시각화된 프로그래밍 언어로 코드를 조합하여 알고리즘을 작성하게 하여 프로그래밍 교육이 이루어진다. 이 연구에서는 국내의 대표적인 EPL 교육 플랫폼인 엔트리 사이트(<https://playentry.org/>)를 이용하여 EPL 교육을 하였다. 이 연구에서 EPL 교육이란 엔트리 사이트를 활용하여 프로그래밍과 알고리즘의 기본 개념을 학습하는 활동을 말한다.

다. 문제해결력

문제해결력이란 지식과 정보를 이용해 문제를 해결하고 검토하여 수정하는 능력을 말한다. 이 연구에서는 문제해결력을 측정하기 위해 학생 수준에 적합한 알고리즘 문제를 제시하였고, 알고리즘 문제를 해결하면서 나타나는 신속성, 정확성, 효율성을 측정하고자 하였다. 이 연구에서 문제해결력이란 정답이 나오기까지 도전한 횟수, 정답에 도달한 시간, 문제 1회당 평균 구상 시간, 오류점검 횟수, 효율적 알고리즘 구상으로 정의하였다.

라. 수업 참여행동

수업 참여행동이란 학생들이 교수·학습과정에 충실히 참여하는 것을 의미한다. 이 연구에서 수업 참여행동에 끼치는 영향이 무엇인지 알아보기 위해 연구 참여자의 특성을 바탕으로 자발적으로 질문하는 행동, 수업에 주의집중하는 행동, 과제회피 행동 3가지 행동을 선정하여 측정하였다.

마. 컴퓨팅 사고력

컴퓨팅 사고력은 컴퓨터 과학의 원리를 활용해 문제를 효율적으로 해결하는 절차적 사고 능력(이신현, 2018)이다. 이 연구에서는 컴퓨팅 사고력의 측정을 위해 선행연구를 바탕으로 비버챌린지 기출문제를 통해 사전-사후 변화를 알아보았으며, 컴퓨팅 사고력을 문제분해, 패턴인식, 추상화, 모델링 및 시뮬레이션, 알고리즘, 평가 총 6가지 하위요소로 알아보았다.

II. 이론적 배경

1. 장애학생과 SW교육

가. SW교육 개념 및 동향

우리나라 정보 교육은 정보통신기술 활용법을 가르치는 교육에서 컴퓨터 과학의 원리를 가르치고 컴퓨팅 사고력을 신장시키는 SW교육으로 변화하고 있다. 컴퓨팅 원리를 이해하고 컴퓨팅 사고력을 갖추는 것은 단순히 좋은 직업을 얻기 위한 것이나 전문 엔지니어를 양성하는 것이 아니라 자신을 둘러싼 일상생활에서 일어나는 삶의 원리를 이해하는 것이다(성정숙, 김현철, 2015; 전수진, 2017). SW교육은 단순히 프로그래밍을 숙달하는 교육이 아닌 컴퓨팅 사고력의 역량을 갖추는 중요한 교육이다. 정보윤리 태도를 가지고 실생활의 문제를 컴퓨팅 사고로 해결하는 데 초점을 두며 이를 위해 문제를 분석하고, 논리적 상관관계를 파악하여 해결 절차를 프로그래밍 언어를 통해 구현해 보는 일련의 과정을 통해 컴퓨팅 사고력을 신장한다(교육부, 2015a). SW교육의 교수·학습 방법은 다양하게 존재하는데 그 중 학생들이 주로 경험하는 SW교육은 언플러그드 활동, EPL 교육, 피지컬 컴퓨팅(physical computing)이다(강두봉, 2019; 문우중 외, 2020; 성영훈, 유승한, 2016).

국내에서 SW교육에 대한 연구 동향(나청수 외, 2018; 이해화, 2018; 황지은, 황성운, 2017)을 살펴보면 SW교육의 연구 주제는 컴퓨팅 사고력, 프로그래밍 교육, 로봇 활용 교육 순으로 많이 이루어지고 있는 것으로 나타났다. 연구 대상은 초등학생과 대학생이 가장 많았고 교사를 대상으로 한 연구물은 상대적으로 적은 것으로 나타나 교육 현장에서 학생을 위한 양질의 SW교육 프로그램 개발 및 설계에 대한 연구자들의 노력이 이루어짐을 알 수 있었고, 학교 현장에서 SW교육이 충실히 적용될 수 있도록 교사 대상 혹은 교사 양성단계부터 SW교육에 대한 지속적인 연구가 필요함을 시사하였다. 특히 언플러그드 교육을 제외한 기타 SW교육 수업유형들은 테크놀로지의 활용능력을 바탕으로 SW교육이 이루어지기 때문에 교사를 대상으로 한 SW교육 연수 프로그램의 연구의 확대가

필요하다고 보았다.

국내 SW교육은 인공지능과 SW를 활용한 새로운 교육적 접근을 시도하고 있다(이정민, 정제영, 2022). 2022 개정 교육과정에서는 초등학교 과정에도 AI교육이 추가되어 SW교육과 AI교육이 연계될 예정이다. 이와 더불어 초등학교 「실과」에 편재되어있는 SW교육의 배정 시간과 교과 관련성이 떨어지기 때문에 초등학교 수준의 정보 교과를 신설할 것을 주장하고 있다(송의성, 임화경, 2021). 하지만 SW교육의 동향에 대한 연구에서는 분석 논문 목록에 장애학생들을 대상으로 한 SW교육 논문은 없었으며, 황지운, 황성운(2017)의 연구에서는 분석과정에서 장애학생 대상 논문이 제외 기준에 들어가 있어 장애학생을 대상으로 한 SW교육의 동향을 알기는 어려웠다.

나. 장애학생의 SW교육

장애학생에게 SW교육이 필요한 이유는 단순히 SW교육이라는 차원을 넘어서 SW교육의 다양한 내용과 활동을 통해 장애학생들의 컴퓨팅 사고력을 향상시키는 데 있다. SW교육을 통해 장애학생이 일상생활 기능을 향상시키고 미래 사회에서 진로 탐색을 할 수 있게 하는 등 독립적이고 효율적인 삶을 살아가는 데 유용한 교육 방법이 될 수 있다(국립특수교육원, 2019). 2022 특수교육통계(국립특수교육원, 2022)에 따르면 특수교육대상자의 72.8%가 일반학교에 재학하고 있는 것으로 나타났다. 이처럼 대다수의 장애학생들은 일반학교에서 통합교육을 받고 있는데 2015 개정 초·중등학교 교육과정에 따라 초등학생은 「실과」, 중학생은 「정보」 교과를 통해 SW교육을 실시하므로 장애학생 역시 SW교육을 접하고 있음을 알 수 있다. 하지만 장애학생이 소속되어 있는 통합학급 담임교사는 장애학생에 대한 교수법이나 자료 부족, 장애학생의 이해 부족으로 통합교육의 목표를 충족시키지 못하는 것으로 나타났으며(강윤아, 신현기, 2019; 김현진, 강옥려, 2008), 일반학교 특수학급에서는 주로 「국어」, 「수학」의 도구교과를 학습하기 때문에 통합학급 담임교사 및 특수교사의 특별한 노력이 없다면 장애학생의 장애 정도와 학습 수준에 적합한 SW교육을 받기 어려운 것이 현실이다. 또한 2015 개정 기본 교육과정(교육부, 2015b)에서는 초등학교 과정의 「실과」 성취 기준을 살펴보면 SW교육 내용이 빠져있다(권순황, 201

8). 중·고등학교의 경우 정규교과목에 포함된 것이 아니라 선택교과인 「정보통신활용」 과목에 SW교육 영역을 포함시켰기 때문에 기본 교육과정을 적용하는 특수학교의 학생들은 학교나 교사의 특별한 관심이 없다면 SW교육을 접하기 어렵다.

한편 급변하는 사회 변화에 따라 교육과정이 개정되었다. 2021년 발표된 2022 개정 교육과정 총론 주요 사항 시안에 따르면 미래 사회 및 환경 변화에 대응하는 교육 내용을 강화하기 위해 여러 교과를 학습하는 데 기반이 되는 언어, 수리, 디지털 소양 등을 기초소양으로 강조하였다(교육부, 2021). 그리고 2022년 12월 22일 국가교육과정 정보센터(ncic.go.kr)를 통해 2022 개정 초·중등학교 교육과정(교육부, 2022a)과 2022 개정 특수교육 교육과정(교육부, 2022b)이 발표되었다. 교육과정은 공통 교육과정, 선택 중심 교육과정, 기본 교육과정으로 나눌 수 있는데 기본 교육과정은 공통 교육과정 및 선택 중심 교육과정을 적용하기 어려운 장애학생을 위해 만들어진 교육과정이다(교육부, 2015b). 장애학생의 개별적인 특성을 고려한 SW교육 내용을 살펴보기 위해 2022 개정 기본 교육과정(이하 2022 개정)과 2015 개정 기본 교육과정(이하 2015 개정)에 제시된 성취 기준을 비교해 보고자 한다.

첫째, 4차 산업혁명 시대의 요구를 반영하여 2022 개정에서는 초등학교 5~6학년군 「실과」 성취 기준에 기본적인 디지털 소양을 함양하기 위한 내용이 신설되었다. 2015 개정 성취 기준에는 SW교육과 관련된 내용이 빠져있으며(권순황, 2018) 정보탐색, 정보통신기기를 이용한 정보 수집 및 인터넷 예절에 관한 내용만 담고 있었다. 2022 개정에서는 언플러그드 놀이 활동을 통해 절차적 사고를 경험하는 내용이 신설되어 특수학교에 재학하는 초등학생들도 미래 사회에 적응할 수 있는 기초적인 디지털 리터러시(digital literacy)를 함양할 수 있도록 하였다.

둘째, 중·고등학교 선택교과인 「정보통신활용」에 제시된 SW교육 내용이 변화되었다. 먼저 중학교 1~3학년의 내용을 살펴보면 2015 개정에서는 문제해결에 필요한 요소를 차례대로 나열하기, 코딩 프로그램의 기초 예제 해결하기, 언플러그드 활동을 통해 문제해결 방법이 소프트웨어로 표현되는 방법 알기에 관한 내용을 제시하였다. 2022 개정에서는 순차, 선택, 반복 구조를 통해 문제해결 과정을 탐색하기, 인공지능에 관심을 가지고 사례 탐색하기에 관한 내용을 제시하였다. 2015 개정의 언플러그드 활동 및 절차적 사고와 관련된 개념이

초등학교 5~6학년으로 이동되었으며, 단순히 코딩 프로그램 해결하기로 진술된 문장에서 순차, 선택, 반복이라는 컴퓨팅 사고력의 알고리즘 요소를 명확히 제시하였고 인공지능에 관한 내용이 신설되었다.

셋째, 고등학교 1~3학년의 SW교육 내용도 변화되었다. 2015 개정에서는 조건을 고려해 문제해결을 위한 단순화된 활동을 순서대로 나열하기, 코딩 프로그램의 기능을 익혀 간단한 문제해결하기, 조건이 포함된 프로그램을 실행하고 오류 점검하기에 대한 내용을 제시하였다. 2022 개정에서는 생활 속 다양한 문제해결 상황을 인식하고 문제해결 절차를 구성하기, 일상생활에서 인공지능 기기를 활용해 편리함 경험하기를 제시하였다. 성취 기준 해설을 살펴보면 2015 개정에서는 문제해결 과정에 대한 학습에서 그림이나 명령어 카드, 즉 언플러그드 활동을 통해 학습하도록 하였으나 2022 개정에서는 EPL을 통해 창의적으로 문제를 해결하도록 변화되었다. 또한 2015 개정에서는 코딩 프로그램의 사용 방법이나 예제 체험하기에 초점을 뒀다면 2022 개정에서는 EPL을 통한 문제해결과 컴퓨팅 사고력 함양에 초점을 두었음을 알 수 있다.

기본 교육과정의 개정 사항에 SW교육 내용이 새롭게 추가되고 심화된 것은 고무적인 일이다. 김동선(2020)의 연구에 따르면 특수교사들은 SW교육에 대해 충분히 인지하지 못하고 있지만 SW교육의 필요성과 유용성에 대해 높게 인식하는 것으로 나타났다. 장애학생의 SW교육이 안정적으로 정착되기 위해 SW교육에 대한 특수교사 대상 연수 및 장애학생 대상 SW교육에 대한 심층적이고 활발한 연구가 이루어져야 할 것이다.

2. SW교육 방법

SW교육의 교수·학습 방법은 다양하게 존재하는데 그 중 학생들이 주로 경험하는 SW교육 유형은 언플러그드 교육, EPL 교육, 피지컬 컴퓨팅 교육이다(강두봉, 2019; 문우종 외, 2020; 성영훈, 유승한, 2016). 언플러그드 교육이란 컴퓨터 과학의 기본 원리인 이진수, 알고리즘적 사고, 프로그래밍 원리를 컴퓨터나 전문 장비를 사용하지 않고 게임이나 놀이를 통해 배울 수 있는 활동을 말한다(김혜란, 최선영, 2019; 장정훈, 김종우, 2016; 전수진, 2017). EPL 교육이란 프로그래밍의 원리를 쉽게 익히기 위해 교육용으로 제작한 프로그래밍 언

어로 프로그래밍 교육을 하는 것을 말한다(서영호 외, 2016). 피지컬 컴퓨팅이란 자신이 만든 코드의 실행을 컴퓨터 화면이 아닌 로봇 같은 물리적 장치로 나타내는 교육 방법이다. 이 3가지 교육이 소프트웨어 기초 교육에 활용되고 있다. 이 중 언플러그드 교육과 EPL 교육은 학교 현장에서 가장 흔히 활용하는 SW교육 유형(강두봉, 2019; 문우중 외, 2020; 성영훈, 유승한, 2016; 신성희, 한경근, 2019)으로 장애학생 대상 SW교육 연구(김소연, 민천식, 2019; 김정일 외, 2012; 신성희, 한경근, 2019; 이현주 외, 2020; 정소영, 2019; 홍다희, 2016)에서 가장 많이 사용된 수업유형이다. 따라서 이 연구에서는 SW교육에서 장애학생 대상으로 연구된 빈도수가 가장 많았던 언플러그드 교육과 EPL 교육의 효과를 비교해 보고자 하며, 각 교육 방법의 개념, 내용, 효과 등을 정리하면 아래와 같다.

가. 언플러그드 교육

1) 언플러그드 교육의 개념

언플러그드 교육은 컴퓨터 과학의 원리와 개념을 컴퓨터 없이 구체물과 놀이 형태의 활동을 통해 학습하는 것이다(김혜란, 최선영, 2019; 최유현, 이승원, 2019; Bell et al., 2009). 캔터베리 대학(University of Canterbury)의 Tim Bell 교수에 의해 개발되었으며 정식명칭은 컴퓨터 과학 언플러그드(computer science unplugged) 프로젝트이다. 이 프로젝트가 시작된 이유는 컴퓨터 과학에 대한 많은 국가들의 관심과 요구가 증가함에도 불구하고 컴퓨터 과학에 관한 학생들의 관심이 감소하는 문제를 겪었다. 완전한 커리큘럼의 부재, 컴퓨터 과학을 가르칠 교사의 부재, 컴퓨터 과학을 이해하지 못하는 관리자로 인해 학생들에게 컴퓨터 과학을 소개하고 제공하는 것이 매우 어려웠다. 이러한 문제를 해결하기 위해 캔터베리 대학의 “컴퓨터 과학 언플러그드” 프로젝트가 시작되었다(Bell et al., 2009).

이 프로젝트는 컴퓨터를 사용하지 않고 컴퓨터 과학의 훌륭한 아이디어를 학생들에게 제공하는 특이한 접근 방식을 취한다. 인터넷 웹사이트(csunplugged.org)를 제작하여 그 안에서 컴퓨터 과학 활동 및 교육을 위한 무료 자료를 제

공한다. 이 자료들은 일반적으로 목표를 달성하기 위한 문제해결과 컴퓨터 과학의 기본 개념을 포함한다. 이 웹사이트는 즉시 인쇄할 수 있는 수업 자료, 영상 자료, 백그라운드 자료, 수업 준비를 위한 체크리스트 등을 포함한 다양한 자료들을 무료로 배포하여 컴퓨터 과학을 쉽게 접근할 수 있게 해준다(Bell et al., 2009; Bell & Vahrenhold, 2018). 이를 통해 교사들은 컴퓨터와 프로그래밍의 완벽한 지식 없이도 컴퓨터 과학을 지도할 수 있게 되었고, 학생들은 컴퓨터 과학을 쉽게 이해하고 흥미로운 교과로 인식하게 하였다.

컴퓨터 작동 원리는 추상적이다. 언플러그드 교육의 내용들은 구체적 조작활동을 통해 추상적인 컴퓨터 작동 원리를 쉽게 학습할 수 있도록 고안된 활동으로 이루어져 있다(한지원, 최연철, 2018). 컴퓨팅 사고력을 길러주기 위해 의도적으로 설계된 활동들로서 컴퓨터 과학이 무엇인지 교육하며 컴퓨팅 사고력에 참여시킨다(김혜란, 최선영, 2019; 최유현, 이승원, 2019; Bell et al., 2009; Bell & Vahrenhold, 2018). 이진수, 알고리즘, 데이터 압축과 같은 컴퓨터 과학 개념을 카드놀이, 문자열, 기타 조작 활동 등으로 컴퓨터 과학의 개념을 소개하고 교육한다(Israel et al., 2015a).

이 프로젝트는 중국, 한국, 일본의 교육자들 사이에서 강력한 협력을 가져왔고, 교육에 관한 아이디어를 공유하고 해당 지역에서 언플러그드를 홍보하기 위한 계획을 개발할 목적으로 2007년과 2008년 중국 우한에서 두차례 워크숍이 열렸다(Bell et al., 2009). 이후 우리나라에서는 15개정 교육과정을 통해 SW 교육이 도입되면서 언플러그드 교육에 대한 연구(강두봉, 2019; 김혜란, 최선영, 2019; 이승원, 최유현, 2019; 장윤재 외, 2011)가 지속적으로 이루어지고 있다.

2) 언플러그드 교육 내용

언플러그드 교육은 컴퓨터가 정보를 표현하는 이진수, 픽셀 단위로 그림을 그려나가는 컴퓨터의 이미지 표현 방식, 반복되는 내용을 줄여서 표현하는 압축의 원리, 정렬, 프로그래밍 언어의 원리 등 정보과학 분야의 전반적인 내용을 담고 있다. 놀이 형식이나 활동으로 구성되어 내용을 쉽게 익힐 수 있고 개인 혹은 그룹 활동을 통해 학습이 이루어진다(교육부, 2017b; 장윤재 외, 2011).

허영(2019)에 따르면 언플러그드 교육은 컴퓨터 과학의 개념, 알고리즘과 프로그래밍의 기본 개념으로 분류할 수 있다고 하였다. 컴퓨터 과학의 개념에서는 이진수, 이미지 표현 등 정보와 자료를 어떻게 표현하고 저장 및 관리하는지에 대해 다루며, 알고리즘과 프로그래밍의 기본 개념에서는 프로그래밍의 근본적인 원리에 해당하는 순차, 반복, 조건들을 다룬다고 하였다.

CS Unplugged 웹사이트에서 제공하는 언플러그드 교육 주제 및 세부 내용은 다음과 같다(출처: csunplugged.org).

첫째, 이진수(binary numbers)이다. 컴퓨터는 숫자를 사용해 정보를 나타낸다. 숫자를 나타내는 가장 간단하고 일반적인 방법은 두 자리 숫자(0과 1)만 사용하는 이진수 시스템이다. 이 주제에서는 이진 시스템이 작동하는 방식과 컴퓨터 데이터가 표현되는 방식을 이해하게 한다.

둘째, 검색을 위한 데이터 구조(data structures for searching)이다. 작업하기 쉽도록 데이터를 구성하는 것은 컴퓨터에서 중요한 기술이다. 이 주제에서는 이진수 검색 트리(binary search tree)와 해시 테이블(hash table)이라는 방법을 통해 컴퓨터가 정보를 저장하고 정보를 찾는 데 빠른 방법은 무엇인지, 컴퓨터에서 일반적으로 사용되는 검색 방법이 무엇인지 탐구한다.

셋째, 오류 감지 및 수정(error detection and correction)이다. 컴퓨터는 불완전하며 디지털 정보를 저장하거나 전송할 때 오류가 생길 수 있다. 이 주제에서는 디지털 장치가 데이터의 오류를 감지하고 수정하는 방법을 탐구한다. 오류를 감지하고 원본 데이터를 복원하기 위해 수정하는 방법, 상대적으로 간단한 아이디어를 통해 디지털 시스템을 안정적으로 만드는 방법을 소개한다.

넷째, 이미지 표현(image representation)이다. 컴퓨터는 숫자로 저장하기 때문에 컴퓨터 이미지는 궁극적으로 컴퓨터 내부에서 0과 1만 사용하여 표현되는 것이다. 숫자를 사용하여 이미지를 만드는 방법, 이미지에 포함할 수 있는 색상 수가 이미지를 저장 할 때 사용되는 비트 수에 따라 결정되는 방법, 이미지를 압축하여 차지하는 방법들을 소개한다.

다섯째, 키드봇(kidbot)이다. 컴퓨터 프로그램을 작성할 때에는 수행할 작업을 계획하고, 코딩하고, 테스트 한 후, 버그를 추적한다. 이 주제에서는 프로그래밍의 요소를 명시적으로 알 수 있도록 학생이 각 요소의 역할을 알아 프로그래밍 언어를 탐구한다. 프로그래밍의 근본적인 원리에 해당하는 순차, 반복, 조건들을 다룬다.

여섯째, 검색 알고리즘(searching algorithms)이다. 컴퓨터는 많은 정보를 처리하므로 효율적인 검색 알고리즘이 필요하다. 이 주제에서는 순차 검색(sequential search)과 이진 검색(binary search)을 학습하여 검색 알고리즘을 이해하고, 올바른 데이터를 안정적으로 찾는 데 걸리는 시간을 비교하는 데 중점을 둔다.

일곱 번째, 네트워크 정렬(sorting networks)이다. 컴퓨터의 속도를 높이는 방법은 더 적은 계산식을 사용하여 프로그램을 작성하거나 여러 대의 컴퓨터가 동일한 작업의 서로 다른 부분을 동시에 작업하는 것이다. 이 주제에서는 병렬 정렬 네트워크(parallel sorting network)를 사용하면 동시 비교를 수행할 수 있는 경우 값이 얼마나 빠른지 탐색한다.

이러한 언플러그드 활동들은 주로 만 5세에서 만 12세 사이의 연령 그룹을 대상으로 한다. 하지만 무조건 해당 연령의 범위로만 제한하지 않으며, 약간의 수정을 통해 청소년이나 성인을 가르치는 데도 사용 가능하다.

3) 언플러그드 교육의 효과

언플러그드 교육은 컴퓨팅 사고력의 기초 개념 습득 및 순차적 사고력을 신장시킬 수 있으며 학생들의 흥미를 유발할 수 있다(이승원, 2019). 또한 추상적인 활동보다는 구체적인 활동을 통해 제시되기 때문에 학생의 흥미, 호기심, 도전 의식을 불러일으켜 자기 주도적 학습 기회가 제공되고, 논리적 사고력을 신장시킨다(김철, 2019; 한선관, 김경신, 2007). 이처럼 언플러그드 교육은 활동, 게임, 노작 등의 수업방식을 통해 학생들의 신체를 많이 활동하게 하는 특징이 있다. 따라서 시스템이 복잡하여 추상적으로 설명할 수 밖에 없는 컴퓨터 작동원리를 이해하는 데 큰 도움이 된다(한병래, 2013).

강두봉(2019) 연구에서는 국내·외 언플러그드 활동과 관련된 시맨틱 네트워크 분석을 실시하였는데 국내 연구에서 언플러그드 교육이 컴퓨터 사고력을 증진시키는 전략으로 사용되었고 그 효과성을 입증하는 연구의 비율이 높게 나타났다. 송한민(2021)의 연구에서는 만 5세 유아를 대상으로 언플러그드 코딩 활동을 실시하였는데 수학적 측면에서 문제해결하기, 다양한 표상양식 활용하기, 사전 경험 활용하여 추론하기의 수학적 경험이 나타났다고 하였다. 이재호, 오

상미(2021)의 연구에서는 초등 저학년을 대상으로 언플러그드 교육을 실시한 결과 컴퓨팅 사고력이 향상되었고 컴퓨팅 사고력 단계 중 데이터 분석과 데이터 표현에서 의미 있는 모습을 보여주었다. 그뿐만 아니라 학생들이 흥미와 호기심을 가지고 교육에 참여하였다고 보고하였다. 김혜란, 최선영(2019)의 연구에서는 초등학교 5학년을 대상으로 언플러그드 컴퓨팅을 활용한 과학 융합 수업을 실시한 결과 자기효능감, 과학적 흥미, 학업성취도에 긍정적인 영향을 준 것으로 나타났다.

이처럼 언플러그드 교육은 여러 연령대의 학습자들에게 컴퓨팅 사고력, 논리적 사고력, 학업성취도, 문제해결력, 추론능력 등 학업 측면에서 긍정적인 성취를 보여주었고, 학습 흥미, 자기 주도적 학습, 자기효능감 등의 정의적 측면에서도 긍정적인 효과를 나타냈음을 알 수 있다.

나. EPL 교육

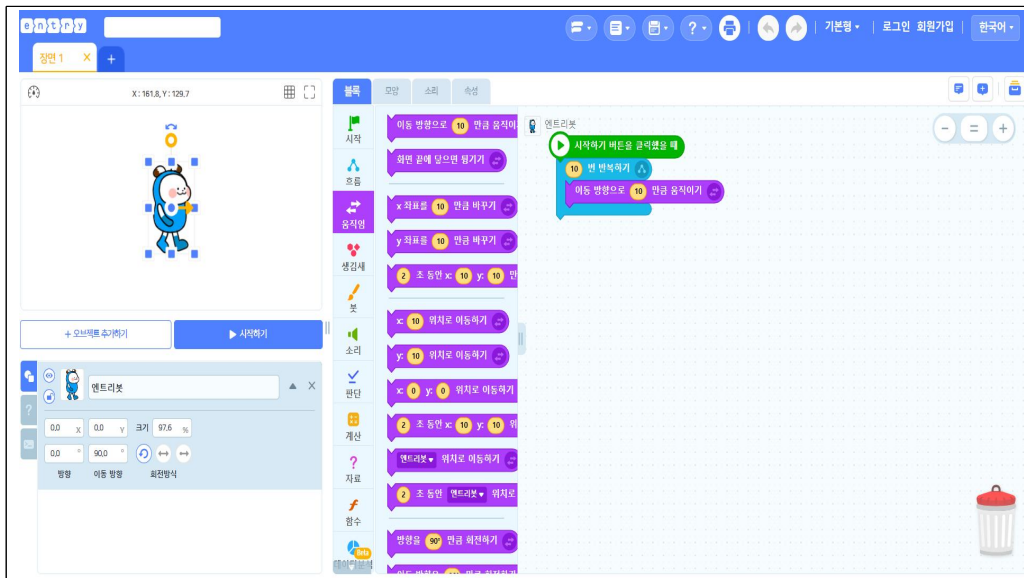
1) EPL 교육의 개념

교육용 프로그래밍 언어는 교육을 목적으로 만들어진 프로그래밍 언어로 학습자가 쉽고 흥미롭게 코딩 언어를 학습할 수 있다. 프로그래밍 교육이 가지는 장점을 부각하고 학습자의 동기와 흥미를 불러일으키기 위해 EPL이 등장하게 되었다(조성한 외, 2008). EPL은 프로그래밍 언어가 시각적 블록 아이콘으로 표현된다. 시각적 아이콘이 블록 모양으로 표현되어 ‘블록코딩’으로도 불린다(임희석, 2017). 각 블록은 컴퓨테이션 요소들을 나타내며 이 블록들을 조립하여 알고리즘을 작성한다. 마치 레고 브릭이나 퍼즐 조각을 맞추는 것처럼 그래픽 블록은 서로 형상이 맞아야 조합이 되고 블록은 의미가 있는 방식으로만 서로 맞도록 설계되어 기존 프로그래밍 언어와 같은 구문 오류를 방지한다(임희석, 2017; Resnick, 2007). 자신이 설계한 알고리즘을 블록을 통해 조합하며 실행화면에서 애니메이션 효과로 즉각적이고 생동감 있게 결과를 확인할 수 있다는 장점이 있다(이은경, 이영준, 2007; 이준형 외, 2018).

C언어, JAVA, Python 등 기존의 프로그래밍 언어는 전문 프로그래머나 전공자들이 사용하는 텍스트 코딩의 종류이다. SW교육의 궁극적인 목적은 컴퓨팅

사고력의 함양인데 전공자들이 사용하는 텍스트 기반 프로그래밍 언어(text based programming language)는 사용 난이도가 높아 초급자가 학습하기에 매우 복잡하고 어려우며 배우기까지 시간이 오래 걸린다는 단점이 있다(오지훈 외, 2021). 이러한 문제점을 해결하기 위해 교육용 프로그래밍 언어가 개발되었다. 국외의 대표적인 EPL 플랫폼은 미국 MIT에서 개발한 스크래치(<http://scratch.mit.edu/>)이며, 국내 플랫폼은 카이스트에서 개발한 엔트리(<https://playentry.org/>)가 대표적이다. 2015 개정 교육과정의 초등학교 5~6학년 실과교과서 6종은 모두 엔트리 플랫폼의 내용으로 구성(이철현, 2019)되어 있다. 엔트리의 기본 화면 구성은 [그림 11-1]과 같다.

[그림 11-1] 엔트리 화면 구성



출처: <https://playentry.org/> (Copyright © NAVER Connect Foundation. Some Rights Reserved.)

프로그래밍이란 주어진 문제를 컴퓨팅 시스템의 도움을 받아 빠르고 정확하게 해결하는 것이다. 제한된 자원과 문제해결에 필요한 핵심 요소들을 고려하여 해결 과정을 수립하고 그것을 컴퓨팅 시스템이 이해할 수 있는 언어로 표현한다. 프로그래밍의 문제해결 과정 중 한 부분이라도 생략되어 있으면 결과를 이끌어 낼 수 없으므로 사고 과정에서 필요한 부분과 불필요한 부분을 구분할 수 있어야 한다. 또한 문제해결 과정에서 도출된 알고리즘을 지속해서 테스트

하고 디버깅함으로써 잘못된 사고 과정을 개선해나가기 때문에 문제해결 능력, 고등 사고력을 향상시킨다(이영준 외, 2014).

프로그래밍 교육은 컴퓨팅 사고력 역량을 강화하고 컴퓨팅 사고력을 활용한 문제해결을 돕는다. 프로그래밍 교육은 컴퓨터 과학의 개념과 원리 습득, 실생활의 복잡한 문제를 창의적으로 해결하는 능력을 길러준다. 컴퓨팅 사고력의 추상화와 자동화 개념을 학습할 수 있으며 컴퓨팅 사고력을 활용한 문제해결 학습에 적절하다. EPL 교육의 목적은 알고리즘적 사고력과 문제해결력, 창의성을 기르는 것이다(오지훈 외, 2021; 윤선희, 2018; 이영준 외, 2014; 이준형 외, 2018). EPL은 이러한 프로그래밍 교육에 매우 적절한 도구로 사용되고 있다. EPL은 컴퓨팅 시스템으로 문제를 해결하기 위한 사고를 촉진시키며, 사고를 실천으로 이어지게 한다. 또한 학습자로 하여금 컴퓨팅 사고력이 무엇인지 자연스럽게 학습하게 한다. EPL 플랫폼은 다양한 튜토리얼을 제공하여 학습자가 수준과 학습 상황을 고려해 선택하여 학습할 수 있어(이영준 외, 2014) 컴퓨팅 과학 교육을 시작하는 학습자에게 매우 적절한 도구가 되고 있다.

2) EPL 교육의 내용

절차적 사고를 구체화하는 것을 프로그래밍이라고 하는데, 문제를 분석하고 이해하며 알고리즘을 설계하고 디버깅 과정을 거치는 것을 말한다(이철현, 2019). EPL 교육에서는 블록을 조립할 때 일반적으로 순차, 선택, 반복 구조를 활용(박준석, 오정석, 2021)하는데 EPL은 하나의 언어로서 이와 관련된 구조문을 학습한 후 알고리즘을 설계하고 문제를 해결해나간다. 이영준 외(2014)는 논리적인 사고 과정을 나타낼 수 있는 도구는 프로그래밍 언어라고 하였으며, 순차, 반복, 조건 등의 컴퓨팅적 개념과 문제해결을 위한 알고리즘 구현 방법을 구체적으로 학습한다고 하였다.

교육부(2015a)에서는 소프트웨어 운영지침을 통해 학교급별로 프로그래밍 교육에 관한 내용, 방법, 평가를 기술하였는데 SW교육과 관련된 내용체계 영역을 3가지로 제시하였다. 첫째, 생활과 소프트웨어, 둘째, 알고리즘과 프로그래밍, 셋째, 컴퓨팅과 문제해결이다. 프로그래밍에서는 알고리즘적 사고가 필수적으로 요구되기 때문에 EPL 교육과 관련된 내용은 둘째, 알고리즘과 프로그래밍

영역이라고 할 수 있다(교육부, 2015a; 심재권, 2019; Futschek, 2006).

교육과정에서 제시한 알고리즘과 프로그래밍 내용은 나선형 교육과정의 형태로 학교급별이 올라갈수록 각 영역의 내용들이 심화 및 확대되어 제시된다(교육부, 2017a). 따라서 초등학교의 SW교육은 SW교육의 기초적인 내용을 담고 있어 장애학생을 위한 EPL 교육에 수정·보완하여 적용하기 적합하다. 따라서 초등학교 SW교육의 알고리즘과 프로그래밍의 교육 내용을 살펴보고자 한다. 초등학교의 ‘알고리즘과 프로그래밍’ 영역의 성취 기준은 <표 II-1>과 같다(교육부, 2015a). 교육부(2015a) SW교육 운영지침에 제시된 알고리즘의 체험 중영역에서 성취해야 할 알고리즘 요소는 순차, 선택, 반복, 정렬, 탐색이며 각 요소의 정의는 다음과 같다.

<표 II-1> ‘알고리즘과 프로그래밍’ 영역의 성취 기준

중영역	내용 요소	성취 기준
문제해결 과정의 체험	문제의 이해와 구조화	제시된 문제를 이해할 수 있다.
		제시된 문제를 단순화할 수 있다.
	문제해결 방법 탐색	문제를 해결하기 위한 방법을 순서에 따라 설명할 수 있다. 제안한 문제해결 방법의 문제점과 개선 방법에 대해 설명할 수 있다.
알고리즘의 체험	알고리즘의 개념	알고리즘의 개념을 이해할 수 있다.
	알고리즘의 체험	순차, 선택, 반복구조를 이용하여 문제해결 절차를 그림이나 기호를 이용하여 표현할 수 있다. 간단한 알고리즘(정렬, 탐색)을 체험활동을 통하여 이해할 수 있다.
프로그래밍의 체험	프로그래밍의 이해	프로그래밍 언어의 기본요소를 알 수 있다.
		주어진 프로그램을 동일하게 만들 수 있다.
	프로그래밍의 체험	주어진 프로그램을 수정하여 자신만의 프로그램을 만들 수 있다. 내가 생각한 간단한 프로그램을 만들 수 있다.

첫째, ‘순차구조’란 어떤 결과를 얻기 위해서 정해진 순서에 따라 차례대로 실행하는 것으로서, 주어진 문제를 컴퓨터로 해결하기 위해 명령어를 순서대로 나열하는 것이다. 코딩 블록은 하나의 블록으로 하나의 이동 동작을 수행하며 각 단계의 처리 순서를 체계적으로 구성해야 한다(박정신, 조석봉, 2012; 심재권, 2019; 이승언, 2019; 장수정 외, 2019).

둘째, ‘선택구조’란 조건 검토 과정을 통해 서로 다른 단계를 수행하는 제어문이다. 만약 ~라면(if else) 블록을 통해 두 가지 조건에 따라 다른 동작을 수행하게 한다(박정신, 조석봉, 2012; 이승언, 2019).

셋째, ‘반복구조’란 문제해결 과정에서 반복되는 부분을 찾고, 반복 명령어를 사용하여 문제를 해결하는 것이다. 반복 기능은 컴퓨터의 신속성과 자동성 활용 측면에서 효율성을 높여준다. 반복 블록을 통해 순차 블록을 반복하는 개수를 조절하여 동작을 수행하게 한다(심재권, 2019; 이승언, 2019; 장수정 외, 2019).

넷째, ‘정렬’이란 특정한 규칙에 따라 대상을 순서대로 배열하는 것이다. 정렬은 숫자뿐만 아니라 순서를 비교할 수 있는 자료를 정렬할 때도 사용된다(장수정 외, 2019).

다섯째, ‘탐색’이란 컴퓨터에서 자료를 찾는 방법을 말한다. 보통 처음부터 순서대로 비교하면서 찾아가는 방법인 선형 탐색을 배운다(장수정 외, 2019).

EPL 교육은 위와 같은 알고리즘의 개념을 학습하고 알고리즘을 체험한 후 EPL 플랫폼을 이용하여 알고리즘 문법들을 사용해 프로그래밍을 이해하고 체험하게 된다.

3) EPL 교육의 효과

프로그래밍은 단순히 컴퓨터 코드를 화면에 나타내는 것이 아니다. 문제를 분해하고 분석하여 절차적 해결 과정을 구상하고 이를 프로그래밍 언어로 표현하여 아이디어를 구체화하는 종합적인 과정이다(이철현, 2015). 논리적인 사고 과정을 실제로 구현하는 강력한 도구는 프로그래밍 언어이므로 프로그래밍의 개념과 과정에 대한 학습은 지속적으로 진행되어야 한다. 이영준 외(2014)는 EPL은 멀티미디어 효과를 통한 직관적인 프로그래밍이 가능하게 하기 때문에 EP

L을 학습해야 한다고 주장하였다. EPL 교육은 학생들이 컴퓨터를 주체적이고 능동적으로 사용할 기회를 주기 때문에 창의적 사고, 수학적 능력, 논리적 사고와 같은 고등 인지 능력을 습득하는 데 매우 긍정적인 역할을 한다(Shin & Park, 2014).

김재경(2018)의 연구에서는 프로그래밍 언어 활용을 중심으로 한 SW교육 연구 동향을 알아보았다. 다양한 프로그래밍 언어 중 EPL이 가장 많이 활용되었으며, 그 중 MIT 공대에서 제작한 스크래치 사이트가 자주 활용되었고, 국내에서 제작한 엔트리 사이트의 활용 사례도 점차 늘어남을 알 수 있었다. 초등 교육과정에서는 학습자의 문제해결력, 성취감, 학습동기, 창의성 향상에 목적을 두고 연구가 진행되었으며 긍정적인 결과를 얻은 것으로 나타났다. 서현석, 정영식(2018)의 연구에서는 교육대학교 1학년 학생을 대상으로 엔트리 프로그래밍 언어를 활용한 협력적 이야기 생성하기 활동을 하였는데 컴퓨팅 사고력 변화에서 사전-사후 검사 결과가 통계적으로 유의하게 높게 나타났다. 윤선희(2018)의 연구에 따르면 전문대학생들을 대상으로 EPL 교육을 한 결과 프로그래밍, 컴퓨팅 사고력, 문제해결력이 향상되었으며 학생들의 수업 만족도 역시 높게 나타났다. 이승원, 이승연(2017)은 초등학교 5~6학년을 대상으로 EPL 중 하나인 스크래치를 통해 공감각 STEAM 교육을 실시하여 기대효과를 알아보았는데 흥미 항목에서 제일 높은 점수가 나왔다. 한선관, 김수환(2010)의 연구에서는 초등학교 3학년과 5학년을 대상으로 EPL을 활용해 수학 문제해결 프로그램을 시행한 결과 동기유발, 사고 과정의 시각화, 메타인지 측면에서 긍정적인 효과가 나타났으며, 수학적 사고력과 수학적 태도의 사후 검사 결과가 높게 나타났다.

EPL 교육은 주로 고학년 학습자를 대상으로 연구가 시행되었음을 알 수 있으며, 학습자들에게 메타인지, 문제해결력, 컴퓨팅 사고력, 수학적 사고력, 프로그래밍 능력 등의 고등 사고력에 긍정적인 변화를 보여주었고, 학습동기, 성취감, 창의성, 흥미, 태도 등 정의적 측면에서도 긍정적인 효과를 나타냈다.

다. 언플러그드 교육과 EPL 교육의 차이

언플러그드 교육과 EPL 교육의 차이를 간단히 정리하면 다음과 같다.

첫째, 교육 시작 시기에 대한 차이이다. 언플러그드 교육은 Tim Bell, Mike Fellow, Ian Witten이 어린 학생들에게도 컴퓨터 없이 컴퓨터 과학을 가르치기 위해 개발하였으며 CS Unplugged 사이트에는 만 5세부터 만 14세까지 학생들이 학습할 수 있는 수준별 자료들이 제공된다. 따라서 언플러그드 교육의 시작 시기에는 이견이 없지만 컴퓨팅 기기를 활용하여 학습하는 EPL 교육의 시작 시기에 대한 논쟁은 있다. White & Sivitanides(2002)는 컴퓨터 과학 과정은 인지 성숙도와 관련이 있으므로 프로그래밍 언어 학습은 전조작기, 구체적 조작기의 인지발달과정에 있는 학생들은 탈진을 경험할 것이라고 예상하였다. 이영준 외(2014)는 아주 어린 학습자는 컴퓨팅 사고력 중 자동화에 대한 학습이 어려울 것이라고 보고, 컴퓨팅 사고력이 본질적으로 추구하는 인지적 영역의 확장을 추구하기보다 컴퓨팅 사고력을 체감할 수 있는 간단한 경험을 제공하는 것이 바람직하다고 보았다. 나원영, 이철현(2016)은 나이가 어린 학습자들은 접근하기 어려운 프로그래밍 언어를 교육하는 데 집중하기보다 언플러그드 수업으로 컴퓨팅 사고력을 학습해야 한다고 보았다. 반면 북유럽 국가들은 취학 전 연령부터 초등학교 저학년까지의 학생들을 위해 EPL 중 하나인 스크래치나 스크래치 주니어를 사용하여 교육하고 있으며 다양한 프로그래밍 활동을 병행하여 수업한다고 알려져 있다. 또한 나라마다 사용하는 EPL 플랫폼은 다르지만, 유아들을 대상으로 태블릿 PC를 사용해 EPL 프로젝트를 수행하는 것으로 나타났다(이재은, 김준수, 2020). 송의성, 임화경(2021)은 다른 나라들에 비해 우리나라의 SW교육 시작 학년이 높아 개정 교육과정에서는 반드시 저학년부턴 SW교육이 이루어져야 한다고 보았다. 이경희(2020)는 컴퓨팅 사고력은 절차적 사고를 통해 분석, 추상화, 문제해결이 이루어지는 종합적 사고 능력이므로 유아에게도 보편적 교육으로 누리과정 내에서 SW교육을 실시해야 한다고 보았다.

둘째, 학습 도구 접근성에 대한 차이이다. 언플러그드 교육은 놀이, 게임 등의 협력 중심의 구체적 조작 활동을 통해 추상적인 원리를 학습할 수 있도록 고안되었으며, 다양한 도구 및 신체만을 이용해 활동이 이루어진다(권은정, 2008; 한지원, 최연철, 2018). 언플러그드 교육 활동에서는 학습지, 실험보고서, 카드, 종이컵, 접시저울 등 쉽게 접할 수 있는 교육자료를 사용한다(한병래, 2013). 언플러그드 교육이 인기를 얻은 이유는 컴퓨팅 프로그래밍 교육을 위한 활동 도구로서 접근이 쉽기 때문이다. 다른 컴퓨팅 프로그래밍 교육들은 시간이나 자원에서 상당한 노력이 필요하다(Bell & Vahrenhold, 2018; Bell et a

I., 2009). EPL 교육은 컴퓨팅 기기를 이용하여 스크래치, 엔트리, 앱 인벤터, code.org 등의 교육 플랫폼 사이트를 접속해 교육이 이루어져(이애화, 2018) 물질적 인프라가 필요하다. 컴퓨터 혹은 태블릿 PC 등의 컴퓨터 기자재가 필요하고, 특정 EPL 교육 플랫폼 사이트를 이용해야 하므로 인터넷 연결이 되는 네트워크 시설이 필요하다(이철현, 2015). 마대성(2021)의 연구에서는 2019 SW교육 선도학교를 대상으로 실태 조사를 실시하였는데, SW교육과 관련하여 중요하다고 생각되는 요소에 대해서는 대부분의 교사들이 SW교육을 위한 학교 시설과 예산, 교구 등이 중요하다고 답을 하였다. 김한성, 전수진(2019)의 연구에서도 SW교육을 위해 디지털 기기 확보, 무선 환경 지원, 관련 예산 지원 항목과 같은 교육환경에 대한 추가적인 지원이 필요하다고 보았다.

셋째, 학습자의 선행 교육과 관련된 차이이다. 언플러그드 교육은 정보과학에 대한 선행지식 및 컴퓨팅 기기에 작동법에 대한 선행 교육이 필요 없어 물리적 제약이 없다(김혜란, 최선영, 2019; 최유현, 이승원, 2019; 한선관, 2010). 반면 EPL 교육은 컴퓨터 마우스 조작법, 드래그 앤 드롭 방식의 조작방법을 알아야 하고, 변수값이나 변수명을 조작해야 할 경우 키보드로 입력할 수 있어야 한다. 또한, 일정 수준의 미디어 문해력이 필요하며, 프로그램 내에서 내가 원하는 기능을 찾고 명령어를 조합하기 위한 교육용 프로그래밍 도구의 지식 습득이 선행되어야 한다(전석주, 2021).

넷째, 교사의 사전 지식과 관련된 차이이다. 언플러그드 교육과 EPL 교육 모두 정확한 교수를 제공하기 위해서는 관련 연수를 받아야 하지만 언플러그드 교육은 CS Unplugged 사이트에서 교육자용 커리큘럼 및 자료를 제공하고 비교적 쉽고 간편한 접근을 통해 학생들을 교수할 수 있다(Yadav et al., 2019). 반면 이소율, 이영준(2017)에 따르면 「정보」교과를 전공한 교사들을 제외한 교사들은 EPL 자체에 대한 지식이 없어 EPL 교육과 관련한 실제적이고 조직화된 연수의 필요성을 느낀다고 하였다. 국내 대표적인 EPL 플랫폼인 엔트리의 경우 선생님께서 회원가입을 하여 반을 생성하기, 학생들의 임시 아이디와 비밀번호를 발급하기, 학생들이 학습해야 할 강의/코스를 추가하기 등(안창호 외, 2018)의 사이트 사용 방법도 익혀야 한다. 한편 문영임 외(2021)의 연구에 따르면 특수교사가 인지하고 있는 프로그래밍 언어는 ‘인지하고 있는 교육 도구가 없음’이 28.10%로 가장 많았으며, 다음으로 엔트리(24.84%), 스크래치(23.53%), 파이썬 57명(18.63%), 앱인벤터 14명(4.58%), 기타(0.33%) 순으로 나

타났다. 연구의 응답자인 특수교사의 상당수가 장애아동 SW교육의 필요성에 대해서는 인지하고 있으나, 대다수가 현재 인지하고 있는 프로그래밍 언어 교육 도구가 없음을 확인할 수 있다.

다섯째, 내용 측면의 차이이다. 언플러그드 교육은 컴퓨터 과학의 원리나 개념을 이해할 수 있는 활동(권은정, 2008; 김혜란, 최선영, 2019)들로만 이루어져 있고 프로그래밍의 과정이 빠져있어(김영기, 한희섭, 2011) 문제해결력과 논리적 사고를 기르기 위한 목적을 달성하기 어렵다는 견해도 있다(한지원, 최연철, 2018). 한편 EPL 교육은 전공자들이 사용하는 텍스트 코딩이 배우기 어렵다는 단점을 통해 만들어진 언어로서 코딩 위주의 프로그래밍 교육이 이루어지고 있다(오지훈 외, 2021). 김필현(2021)은 언플러그드 교육에서는 언플러그드와 컴퓨팅 두 요소를 모두 만족하는 내용으로 구성해야 한다고 보았다. 컴퓨팅 요소는 프로그래밍을 통해 만족시킬 수 있으며 이는 언플러그드 교육의 심화 단계 부분에 해당한다고 보아 언플러그드 교육도 프로그래밍 과정이 포함된

<표 II-2> 두 교육 방법의 차이

영역	언플러그드 교육	EPL 교육
교육 시작 시기	만 5세부터 시작 가능	어린 학습자에게는 부적합하다는 의견도 있으나, 저연령 학습자도 학습 가능하다고 보고 유아를 대상으로 한 연구도 늘어나는 추세임
학습 도구	학습지 및 활동지, 카드, 종이컵, 접시저울 등	컴퓨터, 태블릿 PC 등의 컴퓨팅 기자재
학습자의 선행 교육	선행교육이 필요 없음	컴퓨팅 기기 작동법에 대한 선행 교육 필요
교수자의 사전 지식	CS Unplugged 사이트의 커리큘럼을 제공 받아 쉽고 간편하게 교수 가능	EPL 관련 연수 필요
교육 내용	컴퓨터 과학의 원리나 개념 이해에만 초점을 맞춘다는 견해도 있으나 다수의 연구는 프로그래밍의 기본 개념도 포함된다고 봄	알고리즘과 프로그래밍의 기본 개념

다는 의견도 있다. 허영(2019)은 언플러그드 교육은 컴퓨터 과학의 개념과 알고리즘과 프로그래밍의 기본 개념을 가르치기 때문에 초기 코딩에 활용되고 있다고 하였다. 따라서 언플러그드 교육은 컴퓨터 과학의 원리나 개념만이 아닌 프로그래밍의 기초를 학습할 수 있음을 알 수 있다. 두 교육 방법의 차이를 요약한 표는 <표 II-2>와 같다.

3. 장애학생 대상 SW교육의 효과

가. 문제해결력

1) 지적장애 학생의 문제해결력

문제해결력이란 다양한 사고 방법을 이용해 주어진 문제를 분석하고 파악한 후 적절한 해결 방안을 수립하여 문제를 해결하는 능력을 말한다(허희옥 외, 2011). 21세기 인재 양성을 위해 필요한 역량을 조사한 연구에 따르면 학습자 역량을 신장시키기 위한 기초능력 개발 영역으로 창의적 능력, 문제해결력, 의사소통 능력, 협력, 테크놀로지 리터러시, 예술적 사고가 선정되었다. 이 중 역량의 중요도 판단을 목적으로 한 델파이 조사에서 문제해결력이 가장 중요한 기초능력 개발 영역으로 나타났다(허희옥 외, 2011). 이처럼 문제해결은 교육자들에게 중요한 목표로 여겨져 왔으며, 과학, 수학 및 교육 분야에서 문제해결은 학생들의 일상생활을 위한 주요 초점이자 기본 구성 요소로 언급되고 있다(Tu & Johnson, 1990). 문제해결은 주어진 상황을 고려하여 목표 및 결과에 도달하기 위해 실행하는 인지적 절차이다. 일반적인 문제해결의 과정은 문제 이해, 해결 계획의 수립, 해결 계획의 실행, 반성으로 진행된다.

특수교육의 궁극적인 교육목표는 독립적인 삶을 스스로 영위하는 것이다. 일상생활을 살아가면서 우리는 매일 문제에 부딪히게 되고 그 문제를 해결해야 한다. 일상생활뿐만 아니라 학교생활, 직업 생활에서 문제해결에 대한 능력이 요구되어 그 중요성이 매우 높다(김영표, 신현기, 2008). 문제를 탐색하고 계획하고, 자신의 행동을 검토하고, 결과를 평가하는 문제해결의 단계들은 초인지적 지식에 영향을 받는다(김현진, 2008). 또한, 인지적 과정은 문제 이해,

계획 고안, 계획 실행, 점검의 4가지 단계를 거친다. 장애를 가진 학생들은 인지 전략을 사용하는 능력이 부족하여 문제해결력에도 어려움을 가지므로 문제를 해결할 수 있는 인지 전략을 개별적으로 지도해야 한다(송준만 외, 2012).

2) SW교육과 문제해결력

현세대 학습자들은 모든 것이 융합되고 초연결되는 디지털 시대에 살고 있다. 소프트웨어 중심으로 돌아가는 사회에 살아가기 위해서는 컴퓨터 과학 원리를 이해하고 이를 통해 문제를 해결할 수 있는 역량이 필요하다(성정숙, 김현철, 2015). 정보과학적인 사고는 정보 처리 과정과 방식을 이해한 후 다시 컴퓨터에게 지시하는 순환적 사고이다. 따라서 문제를 해결하는 과정에서 정보과학적 원리를 적용해 알고리즘을 설계하고 구현한 후 결과를 검토한다. Tu & Johnson(1990)은 프로그래밍 기술과 문제해결력 간의 관계에 대해 연구하였다. 대학생들을 대상으로 컴퓨터 프로그래밍 과정을 실시하고 문제해결 능력에 대해 사전-사후 검사를 시행한 결과 컴퓨터 프로그래밍 과정을 성공적으로 이수한 학생들은 문제 기술의 전반적인 과정에서 상당한 향상을 경험한 것으로 나타났다. 또한 문제해결 평가의 논리적 추론 영역에서 상당한 향상을 보여주었다. 김갑수(2010)는 프로그래밍 언어 교육을 통해 문제해결, 문제를 찾는 능력, 문제 관리 능력이 생긴다고 보았다. 학생들이 프로그래밍을 통해 문제를 해결해 나가면서 에러를 찾고 이를 수정하는 과정을 통해 구체적인 문제해결 과정을 관리할 수 있는 능력이 길러진다고 본 것이다. Gomes & Mendes(2007)는 학생들이 알고리즘을 만드는 것을 모르기 때문에 프로그래밍이나 문제해결 능력에 어려움을 보인다고 하였다. 따라서 문제해결 능력을 높여야 프로그래밍 능력의 향상을 기대할 수 있다고 하였다.

SW교육에서의 문제해결은 일반적인 문제해결과 다른 해결 과정을 갖는다(장재득, 주길홍, 2015). SW교육에서 문제해결은 답을 원하는 것보다 답을 해결하는 방법을 찾길 원한다(한병래, 2013). 주어진 문제를 해결하기 위해서는 절차를 만들어내는 능력인 알고리즘적 사고가 필요하다. 알고리즘적 사고를 위해 컴퓨팅의 기본 절차인 순차구조, 반복구조, 선택구조에 대한 이해는 필수적으로 요구되며 반드시 가르쳐야 할 요소로 강조하고 있다. 학생들은 주어진 문제

에 대해 스스로 생각하며 문제해결 절차에 따라 알고리즘을 설계하도록 교육과 평가가 이루어진다(심재권, 2019; 정상목, 2020).

알고리즘은 SW교육 문제해결에 필요한 요소이며 SW교육을 통해 얻을 수 있는 중요한 역량이다. Futschek(2006)는 그의 연구에서 알고리즘을 주어진 문제를 정확하게 해결하기 위해 정의된 명령어로 이루어진 풀이 방법이라고 정의하며 알고리즘적 사고를 첫째, 주어진 문제를 분석하는 능력, 둘째, 문제를 정확하게 구체화하는 능력, 셋째, 주어진 문제에 적합한 기본 행동을 찾는 능력, 넷째, 기본 동작을 사용하여 주어진 문제에 대한 올바른 알고리즘을 구성하는 능력, 다섯째, 가능한 일반적인 경우에 대해 생각하는 능력, 마지막으로 알고리즘의 효율성을 향상시키는 능력이라고 하였다. 장재득, 주길홍(2015)은 알고리즘의 조건을 입력, 출력, 명확성, 유한성, 효율성으로 보았다. 또한 김세용(2002)의 연구에서는 알고리즘을 정확성(correctness), 수행횟수(amount of work done), 간단성(simplicity), 최적성(optimality)으로 분석하였다.

알고리즘을 설계하고 코딩하는 과정은 정해진 문법 규칙에 따라 문제해결을 위한 논리를 구성하는 작업이다. 이 작업에서 끊임없는 디버깅(debugging)은 필수요소 중 하나이다. 디버깅이란 자신이 프로그래밍한 내용을 확인하고 오류를 수정하는 것이다. 스스로 디버깅하여 결과를 예상하고 예상한 결과가 나오지 않았을 때 그 결과가 왜 그렇게 되는지 자기반성을 경험하게 된다. 이 과정을 통해 오류는 실패가 아닌 더 나은 결과를 내기 위한 발판이 된다는 것을 경험한다(노지예, 이정민, 2018; 이영준 외, 2014; 이재호, 차근민, 2020). 조성환 외(2008)에 따르면 프로그래밍과 관련된 문제해결 과정에서 디버깅과 같은 오류 수정의 과정을 필수요소로 보았으며, 남수정(2020)은 그의 연구에서 문제해결력을 측정할 때 학생들이 문제와 해답의 일치성을 확인하고 오류를 수정하였는지 확인하였다.

위 내용을 종합하면 SW교육에서 문제해결은 알고리즘적 사고를 통해 문제를 해결하는 것이며, 알고리즘의 중요 요소는 정확성, 효율성, 디버깅임을 알 수 있다. 정확성은 올바르게 알고리즘이 세워졌는지, 효율성은 가능한 적은 자원을 사용해 효율적으로 논리 구조를 표현하는 것을 말하며, 디버깅은 자신이 세운 알고리즘의 오류를 점검하여 수정하는 것으로서 이 세 가지는 SW교육 문제해결에 필수적인 요소라고 볼 수 있다.

나. 수업 참여

1) 수업 참여행동의 개념

수업 참여는 그 자체로 중요한 학업의 결과이다. 수업 참여는 학습 성과를 향상시키고 학업 능력에 대한 긍정적인 기대를 갖게 한다. 수업 참여는 교사와 학생 간 지지적인 상호 반응을 이끌어낸다. 학생들의 수업 참여를 통해 교사는 더 많은 동기부여를 할 수 있는 지원을 제공한다. 학업 활동에 대한 참여의 질이 높아진다면 장기적인 학습, 사회화 및 발달에 긍정적인 영향을 미칠 것이다 (Furrer & Skinner, 2003). 수업 참여행동이란 수업에서 이루어지는 과정에서 관찰되는 행동(변관석, 2016)으로 각 연구마다 연구참여자의 특성에 맞게 정의와 요소를 달리하였다.

Rhode 등(1983)은 적절한 교실 행동을 주의집중, 과제 수행, 자진해서 말하기, 크게 읽기, 질문에 답하기, 교사에게 질문하기로 보았다. 김남희(2011)는 수업 참여행동을 교육적으로 의도된 활동에서 학생들이 전념하는 노력으로 정의하고 주의집중, 과제 수행, 활동 참여, 발표하기 요소를 측정하였다. 유환조(2015)의 연구에서는 수업 참여행동을 수업 상황에서 나타나는 교사-학생, 학생-학생 간의 기능적이고 유기적인 행동을 보고, 수업 참여행동을 담임교사와의 면담 및 기능평가를 통해 연구참여자에게 가장 요구되는 행동들인 집중하기, 지시 따르기, 과제 수행하기로 선정하였다. 변관석(2016)은 발달장애 학생의 수업 참여행동을 지시 따르기, 응답하기, 과제 수행하기로 구분하였으며 수업을 촬영한 장면을 분석하며 행동 발생률을 측정하였다. 서현주, 박은혜(2018)는 스위치를 활용한 AAC 중재가 지체·중복장애 고등학생의 수업 참여행동에 미치는 영향을 알아보았는데, 이 연구에서 수업 참여행동은 반응과 주의집중으로 분류하였으며 수업을 촬영한 장면을 분석하여 측정하였다. 반응에서는 스위치, 발화, 몸짓 반응하기로 구분하여 교사의 질문에 10초 이내 답변하는 유형을 측정하였고, 주의집중은 3초 이상 교사를 바라보거나 수업에 참여한 행동을 측정하였다. 이상철(2012)은 지적장애 학생의 수업 참여행동을 주의집중 행동, 수업활동 참여, 과제 수행하기로 제시하였다. 주의집중 행동은 교사의 언어적 촉구 없이 주목하는 행동, 교사의 언어적 촉구를 받아 또래의 발표를 바라보는

행동, 수업자료를 바라보는 행동으로 정의하고 학생의 행동을 측정하였다. 수업활동 참여는 교사의 언어적 단서로 자발적으로 참여하는 행동, 스스로 학습지와 수업자료를 정리하는 행동으로 정의하였다. 과제 수행하기는 교사가 제시하는 과제에 시선과 관심을 주고 교사의 언어적 단서를 통해 수행하는 것으로 정의하였다. 수업을 촬영한 장면을 분석하여 행동 발생률을 측정하여 효과를 알아보았다. 권호진(2012)은 수업참여도를 수업준비 활동, 주의집중 행동, 과제 수행행동, 적절한 반응 행동 4가지로 분류하였으며 총 14문항을 제작하여 관찰 교사와 연구참여자에게 점검표에 평정하도록 하였다.

이처럼 각 연구에서는 연구 주제나 학생의 특성에 맞추어 수업 참여행동을 다양하게 제시하였다. 이 연구에서는 선행연구 및 학생 특성을 바탕으로 수업 참여행동을 자발적 질문하기, 주의집중 행동, 과제회피 행동으로 정의하였다.

2) 지적장애 학생의 수업 참여행동

(가) 자발적 질문

자발적 질문은 자연스러운 상황에서 질문자가 알고 싶은 것을 묻는 발화 형태로 만 2~3세에 자발적으로 왕성한 질문 활동을 시작하는 것으로 나타났다(주현정, 이은미, 2009). 질문이라는 행동은 다양하고 복잡한 인지 능력의 조정을 필요로 하며 배움에 대한 촉매제가 될 수 있는 능력이다. 학습자의 질문은 주체적인 학습을 가능하게 하며 지식 성장의 향상을 결정하는 중요한 역할을 한다. 질문은 인지적 불일치나 무지를 각성하고 언어적으로 구조화되었을 때나 문제를 해결하고자 하는 의지가 있을 때 질문을 하게 된다. 즉 해결책을 알고 싶은 필요와 의지가 동반될 때 개인의 지적인 능력을 성장시키는 질문이라는 행동이 나타난다. ‘질문 있는 교실’ 혹은 ‘거꾸로 학습’ 등을 통해 학생들이 제기하는 질문에 의해 수업의 방향이나 목적이 결정되며, 이를 통해 수업에서 학생의 질문을 끌어내기 위한 관심이 조금씩 생겨나고 있다(김명숙, 2015). 하지만 학생들이 보이는 탐색 행동과 비교해 질문에 대한 연구는 상대적으로 드물다(양미경, 2007; Ronfard et al., 2018).

장애학생과 관련한 자발적 질문에 관한 연구는 주로 자폐성장애 학생 대상으

로 이루어져 왔다. 이들이 보이는 무발화 혹은 의사소통의 질적 결함으로 자발적 발화를 이끌어내기 위해 자발적 질문에 대한 언어적인 중재 연구가 이루어져 왔다(길한아, 손승현, 2018; 김경희, 2015; 김미영, 김은경, 2012; 최수영 외, 2021; 홍성희, 2003). 자폐성장애 외에 장애학생을 대상으로 실시된 자발적 질문에 대한 연구는 주로 수업 참여행동의 중 하나로 분류되어 연구가 이루어짐을 알 수 있었다. 김병곤(2019)은 지적장애 학생을 대상으로 칸 아카데미 수학 학습 프로그램을 통해 과제 수행행동에 미치는 영향을 알아보았으며 이 연구에서는 과제 수행행동 4가지 중 하나로 질문하기를 선정하였다. 연구 결과 모든 학생들이 질문하기 행동이 늘어난 것을 알 수 있었다. 김태용, 최하영(2013)은 장점강화 활동이 지적장애 초등학생의 수업 참여행동에 미치는 영향을 연구하였다. 연구참여자 특성을 ‘질문을 하는 경우가 없고, 주로 수업과 관련 없는 질문을 한다.’ 라고 제시하며 수업 참여행동 중 자발적 행동을 수업 활동에 관련하여 질문하기, 수업이나 학습활동 관련하여 스스로 먼저 질문하기로 정의하고 행동을 측정하였고 연구 결과 자발적 행동의 발생 변화가 긍정적으로 변화되었다. 고정현, 이한얼(2015) 연구에서는 지적장애 학생에게 오르프 교수법 중재를 통해 음악과 수업참여도를 분석하였다. 수업참여도 중 하나로 ‘도움 요청 또는 질문’을 선정하였고, 이 영역을 학습주제와 관련된 문제를 해결하기 위한 도움 요청 또는 질문이라고 정의하였다. 수업참여도를 측정한 결과 수업참여도가 상승하였음을 알 수 있었다. 김완숙, 방명애(2011)는 루브릭 참조 자기평가를 통해 장애학생의 수업 참여행동의 변화를 연구하였다. 이 연구에서는 장애학생의 수업 참여행동을 IEP(개별화교육계획) 학습목표와 연계시켜야 한다고 보고, 연구참여자인 지적장애 학생의 수업 참여행동 중 하나로 자발적 참여행동을 선정하였다. 자발적 참여행동은 교사에게 질문하기 위해 손들기, 친구에게 과제에 대해 질문하기, 수업내용과 관련된 내용에 대해 교사에게 질문하기 등으로 정의하고 행동을 측정하였다. 연구 결과 수업 참여행동 발생률이 향상됨을 알 수 있었다. 이선영, 강옥려(2021)는 수업 참여행동을 3가지 유형으로 나누었으며 그 중에 자발적 행동으로 ‘수업과 관련된 질문을 한다’로 정의하여 행동을 측정하였다. 차재경, 김진호(2014)는 지적장애 학생에게 긍정적 행동지원을 제공하여 수업 참여행동과 방해행동에 미치는 영향을 연구하였다. 교사의 면담 및 사전 조사를 통해 연구참여자가 가장 필요한 수업 참여행동을 3가지 선정하였는데 그 중에 하나로 자발적 행동을 선정하였으며 여

러 가지 자발적 행동 중 교사에게 수업과 관련된 질문을 하는 행동을 측정하였다. 연구 결과 수업 참여행동을 사전-사후로 비교했을 때 사후에 20%가 상승하였음을 알 수 있었다.

위의 내용을 종합해 볼 때 지적장애 학생의 질문에 관한 연구는 수업 참여행동의 요소 중 하나로 선정되어 측정되고 있었으며, 연구 결과 수업 참여행동이 긍정적으로 변화됨을 통해 간접적으로 질문하기 능력이 신장되었다는 것을 알 수 있었다. 이 연구에서도 선행연구 및 연구참여자의 특성을 고려하여 수업 참여행동의 하나로 자발적 질문하기를 선정하였다.

(나) 주의집중

주의란 동시에 제시된 자극 중에 필요한 특정 자극에 선택적 반응을 지속하고 그 외 자극에는 반응하지 않는 행동인 선택적 주의집중(selective attention)과 방해 자극을 무시하며 집중된 주의를 유지하는 주의 유지(sustained attention)로 구성된다. 주의집중 능력은 일상생활 및 학습 전반에 기본이 되는 능력이다. 새로운 과제를 학습할 때 필요한 자극에 집중하고 무관한 자극을 배제하지 않는다면 과제를 이해하지 못하여 바르게 수행하지 못한다. 지적장애 학생들은 자극에 의식을 집중하는 인지 활동 능력이 약하여 주의집중 시간이 짧고 적절한 곳에 주의를 기울이기보다 필요 없는 자극에 주의를 기울인다. 이러한 주의 문제는 단기기억에도 영향을 주고 나아가 학습 전반에 영향을 끼치게 된다. 주의집중의 결함은 지정된 과제나 교사에게 올바른 집중이 되지 않으며 지시 따르기에 문제도 문제가 되어 학습효과가 크게 약화 된다. 하지만 주의집중 능력은 연습이나 환경 자극을 통해 개선될 수 있다고 보았다. 따라서 주의집중은 지적장애 학생의 교수전략에 고려되어야 할 요소로 보았다(송준만 외, 2012; 이소현, 박은혜, 2011; 전해진 외, 2016; 조미현, 2012).

주의집중은 학생들의 학업과 직결되는 부분인 만큼 여러 연구에서 수업 참여행동 요소 안에 주의집중 행동을 선정하였다(권효진, 2012; 김미화 외, 2020; 서민영, 정은희, 2021; 서현주, 박은혜, 2018; 이상철, 2012; 이선영, 강옥려, 2021). 이선영, 강옥려(2021)는 지적장애 초등학생을 대상으로 계산기를 활용한 화폐교육을 실시하여 수업참여도에 미치는 효과를 보았는데 선행연구를 토

대로 수업참여도 행동을 주의집중 행동을 포함한 3가지 행동으로 정의하고 측정하였으며 연구 결과 수업참여도는 기초선에 비하여 높은 수행률을 보인 것으로 나타났다. 권효진(2012)은 지적장애 중학생을 대상으로 보편적 학습설계를 적용한 과학 수업을 실시하여 수업참여도에 미치는 영향을 연구하였다. 수업참여도는 수업장면과 밀접하게 관련된 공통 요소들을 추출하여 4가지 행동으로 선정하였는데 그 중 주의집중 행동도 선정되었다. 연구결과 실험집단의 수업참여도 4가지 행동이 모두 증가하였으며, 그 중 주의집중 행동이 가장 큰 증가율을 보였다. 이상철(2012)은 사회과 정착수업을 통해 지적장애 학생의 주의집중에 미치는 영향을 알아보았다. 주의집중에 문제를 보이는 학생들을 대상으로 중재를 실시하였으며 연구참여자 모두 주의집중 행동이 증가한 것을 알 수 있었다. 서민영, 정은희(2021)의 연구에서는 스토리텔링을 적용한 수학 수업이 지적장애 고등학생의 수업 참여행동에 미치는 영향을 미치는지 알아보았다. 수업 참여행동 중 하나로 연구자의 설명에 집중하기의 행동을 포함하였으며 수업이 진행될수록 수업 참여행동의 증가율을 보였다.

위의 내용을 종합해 볼 때 지적장애 학생들은 개별적 요구를 고려한 다양한 중재를 통해 긍정적이고 적극적인 수업 참여가 이루어졌고 이는 주의집중의 측정을 통해 그 효과를 알 수 있었다. 이 연구에서도 학생의 흥미를 자극한다(Ayer, 2018; Hufad et al., 2021)고 알려져있는 언플러그드 교육과 EPL 교육을 실시하여 학생들의 주의집중 행동에 향상이 있는지 알아보고 수업유형 간 차이를 비교해 보고자 한다.

(다) 과제회피 행동

부적절한 행동은 여러 가지 환경 안에서 자신이 원하는 변화를 가져온다. Alberto & Troutman(2010)은 그의 저서에서 학생들이 보이는 행동의 여섯 가지 기능을 제시하였다. 행동의 기능은 첫째, 관심을 얻기 위해 하는 행동, 둘째, 유형의 무언가를 얻기 위해 하는 행동, 셋째, 감각자극을 얻기 위해 하는 행동, 넷째, 관심이나 상호작용을 도피하기 위해 하는 행동, 다섯째, 과제에서 도피하기 위해 하는 행동, 여섯째, 고통스럽거나 불편한 내적 자극에서 도피하기 위해 하는 행동이다. 이 중 과제를 도피하는 행동은 과제를 하는 것에서 벗

어나기 위해 하는 행동으로서 과제가 너무 어렵거나 과제가 너무 쉬울 때 일어나는 행동이다. 만일 학생이 도피를 요청할 의사소통 기술이나 사회적 기술을 가지지 못했다면 사회적으로 부적합한 부적절한 행동을 할 가능성이 크다. 그러한 행동들이 원하는 대로 바뀌게 되면 그 결과의 성취는 강화되며 학생이 부적절한 행동으로 과제에서 도피할 가능성과 비율을 증가시킨다(Grey & Hastings, 2005).

지적장애 학생은 자신의 능력이 평가되는 수업 상황을 불안해하고 위협적으로 인식하여 자신의 무능력이 드러날 수 있는 학습 상황에서 회피적인 태도를 보인다(정희정, 2006). 또한 학습된 무기력(learned helplessness)으로 인해 과제를 회피하기도 하는데, 학습된 무기력이란 누적된 실패경험으로 인해 학업에 대한 의지를 상실하는 것을 말한다. 한경화, 김영옥(2016)에 따르면 경도장애 학생은 주어진 문제를 집중해서 풀기보다 대충 풀고 포기를 쉽게 하는 경향을 보인다고 하였다. 지적장애 학생들은 학습된 무기력으로 학습동기가 낮은 모습을 많이 보이기 때문에 수준에 적합한 도전 과제를 제시하여 성공을 경험하게 하는 일이 중요하며 시간이 다소 걸리더라도 스스로 해결할 기회를 많이 제공해야 한다(이소현, 박은혜, 2011).

과제회피 행동은 그 자체로 도전적 행동이나 문제행동으로 분류되어 중재연구(김정일, 권미영, 2014; 양지숙, 허유성, 2020; 이명주 외, 2011; 정지은, 양문봉, 2021)가 이루어지기도 하지만 수업 참여행동의 중 하나로 분류되어 연구(고정현, 이한얼, 2015; 임해주, 2017)가 이루어지기도 한다. 고정현, 이한얼(2015)은 오르프 교수법을 통해 지적장애 학생의 수업참여도에 미치는 영향을 알아보았는데, 수업참여도를 수업 참여행동과 비참여행동으로 분류하고 비참여행동을 학습주제 외 행동과 반응없음에 대한 행동으로 정의하여 연구를 실시하였다. 연구 결과 비참여행동에서 빈도와 비율이 감소되었다. 임해주(2017)는 경도 지적장애 학생을 대상으로 기술·가정 교과와 교수적 수정을 통해 수업참여도에 미치는 영향을 연구하였다. 수업참여도를 수업 참여행동과 수업일탈행동으로 분류하고 수업일탈행동을 수업에 무관한 행동하기, 자리 이탈하기, 손장난하기, 과제회피하기로 정의하였다. 연구 결과 과제회피 행동이 감소하였음을 알 수 있었다.

위를 종합하면 과제회피 행동은 과제에 대한 난이도 조절에 실패했을 때, 평가되는 상황을 피하고 싶을 때 일어나는 행동이며 교수적 수정이나 학생의 흥

미를 고려한 중재를 제시했을 때 과제회피 행동이 감소하는 것을 알 수 있다. SW교육은 알고리즘을 설계하고 평가하며 수정하는 과정이 반복된다. 계속되는 평가상황에서 학생들은 과제회피를 하지 않고 지속적으로 수업에 참여하는지 또한 두 중재 간 차이가 있는지 알아보기 위해 이 연구에서는 선행연구를 참고하여 수업 참여행동 중 하나로 과제회피 행동을 포함하여 수업유형 간 차이를 비교해 보고자 한다.

다. 컴퓨팅 사고력

1) 컴퓨팅 사고력의 개념 및 의의

컴퓨팅 사고력(computational thinking)이라는 용어는 1980년도에 Seymour P apert에 의해 처음 언급되었지만, 2006년 Jeanette Wing이 용어를 정교화하고 대중화하여 국제 사회의 관심을 끌게 하였다. 이후로 교육 분야에서 컴퓨팅 사고력에 대한 많은 연구가 수행되었다(오지훈 외, 2021; 이철현, 2015; Lockwood & Mooney, 2017). Wing은 컴퓨팅 사고력을 ‘컴퓨팅의 기본 개념을 바탕으로 문제해결, 시스템 설계, 인간 행동 이해에 접근하는 것이며, 컴퓨터(인간 또는 기계)가 효과적으로 수행하는 방법으로 문제를 공식화하고 해결방법을 표현하는 것과 관련된 사고 과정’이라고 정의하였다. 또한 학생들의 기본 능력인 읽기, 쓰기, 연산 능력에 컴퓨팅 사고력을 추가해야 한다고 하며 21세기를 살아가는 모든 이들이 가져야 할 기초 기술이 될 것이라고 하였다(Barr et al., 2011; Wing, 2008; Wing, 2006). 유지원(2022)은 컴퓨팅 사고력이란 컴퓨터 과학의 원리를 활용해 문제를 발견하고 이해하며, 해결 방향을 설정한 후 해결안을 개발해 효과적으로 문제를 해결하는 절차적 사고라고 하였으며, 김민정 외(2017)는 컴퓨팅 사고력이란 컴퓨터 과학에 대한 지식을 기반으로 문제를 해결하는 힘이라고 정의하였다. 이철현(2015)은 컴퓨팅 사고력을 문제의 구조를 분석 및 분해하여 공통적 패턴을 찾아 추상화하고, 논리적인 문제해결 절차를 알고리즘으로 구안하며, 문제해결 과정을 프로그래밍함으로써 해결책을 산출하는 종합적인 문제해결력으로 정의하였다. 이영준 외(2014)는 컴퓨팅 사고력을 컴퓨팅 시스템의 역량을 활용하여 해결해야 할 문제를 효율적이고 효과적으로 해결할

수 있는 절차적 사고 능력이라고 하였다. 학자마다 정의는 다르지만 공통으로 들어가는 중요 개념은 문제해결이라고 할 수 있다.

컴퓨팅 사고력이란 컴퓨터로 문제를 해결하는 것이 아니라 컴퓨터 과학의 개념을 이용해 실제 문제를 해결하는 것이다. 컴퓨팅 사고력을 발휘하는 사람은 문제에 대한 다양한 접근 방식을 고려한다. 회귀적 접근 방식, 순차 및 동시 접근 방식, 혹은 유용한 추상화를 제시할 수 있다. 문제가 반드시 컴퓨터와 관련된 것은 아니나 컴퓨터 과학의 개념을 적용하여 실제 문제를 해결하게 하는 것이다(Bell & Vahrenhold, 2018). 컴퓨팅 사고력의 2가지 주요 개념은 추상화(abstraction)와 알고리즘(algorithm)이다. 추상화는 복잡한 내용을 간소하게 표현함으로써 효율적인 해결방법을 설계할 수 있게 한다. 알고리즘은 컴퓨터가 수행할 수 있는 프로세스를 만들 수 있게 해준다(서영호 외, 2016; 임희석, 2017).

이영준 외(2014)는 컴퓨팅 사고력은 실생활에서부터 학문 분야까지, 간단한 문제에서 복잡한 문제까지 직면한 문제들을 해결하는 데 필요한 능력으로 보았다. 컴퓨팅 사고력의 본질은 복잡한 문제를 보다 친숙하고 관리 가능한 하위 문제로 분해(문제 분해)하고, 일련의 단계(알고리즘)를 사용하여 문제를 해결하며, 솔루션이 유사한 문제로 어떻게 전환되는지 검토(추상화)한 후 최종적으로 결정하는 것을 포함한다. 마지막으로 컴퓨터가 이러한 문제를 보다 효율적으로 해결하는 데 도움이 되는지 확인(자동화)한다(Yadav et al., 2016). 컴퓨팅 사고력이 증진되면 복잡성 처리에 대한 자신감, 어려운 문제를 해결하려는 끈기, 모호성에 대한 허용심, 개방형 문제를 다루는 능력, 공동의 목표 또는 솔루션을 해결하고 달성하기 위해 다른 사람과 의사소통하고 협력하는 능력이 신장한다고 보았다(Barr et al., 2011).

2) 장애학생의 컴퓨팅 사고력

컴퓨터 과학 교육 분야에서는 학령기 학생들의 컴퓨팅 사고력을 신장시키기 위한 학습 기회를 확대하기 위해 전례 없는 노력을 기울이고 있지만 그 노력에 장애학생은 제외되는 것이 현실이다(Wille et al., 2017). González-González et al.(2019)은 장애학생들은 장애가 없는 학생보다 더 느리고 다르게 배우기

때문에 동일한 기술을 습득하는 데 많은 시간이 걸리지만, 장애학생이 수학적 사고를 습득하는 것과 같은 방식으로 컴퓨팅 사고를 습득할 수 있다고 보았다.

장애학생의 컴퓨팅 사고력에 관한 연구는 제한적이지만(Bouck & Yadav, 2022) 장애학생의 컴퓨팅 사고력 교육에서 즉각적인 피드백과 명시적 교수가 효과적임을 발견하였고(Israel et al., 2015b), 지적장애 초등학생에게 코딩을 성공적으로 가르친 연구도 있었다(Taylor, 2018; Taylor et al., 2017). Bouck & Yadav(2022)는 장애학생의 수학 학습을 지원하기 위해 컴퓨팅 사고력의 활용을 제안하였다. 수학에서 컴퓨팅 사고력을 구현하기 위해 언플러그드 및 플러그 활동을 제시하였다. 먼저 언플러그드 활동에서는 이진수 학습을 통해 추상화 개념을 교수할 수 있다고 하였으며, 플러그 활동에서는 수학 이야기나 수학 문제를 설명하기 위해 스크래치와 같은 코딩 프로그램을 사용할 것을 제안하였다. 특히 플러그형 활동은 코딩 프로그램이나 코딩 로봇과 같은 프로그래밍 기술을 교육에 통합시킬 기회를 제공할 수 있다고 보았으며, 장애학생도 컴퓨팅 사고력이나 컴퓨팅 과학에 접근할 권리가 있으므로 의도적으로 수학 학습에 통합함으로써 학생들에게 접근 기회를 늘려야 한다고 하였다.

Israel et al.(2018)에 따르면 장애학생에게 지원을 제공했을 때 컴퓨팅 사고력에 긍정적인 결과를 보이는 연구 결과에도 불구하고 장애학생에 적합한 컴퓨터 과학 교수전략을 교수할 교사 자원이 거의 없으므로 장애학생의 요구를 충족하는 데 어려움을 겪는다고 하였다. 따라서 장애학생들을 컴퓨터 과학 활동에 포함시키기 위한 교수전략을 교사에게 제공하는 것이 중요하고 필요하다고 보았다.

4. 장애학생 대상 SW교육 연구동향 분석

장애학생을 대상으로 SW교육을 실시한 중재 연구는 많지는 않지만 2012년부터(김동일 외, 2012; 김정일 외, 2012) 꾸준히 실시됐다. 특히 2015 개정 교육 과정을 통해 SW교육이 도입되기 이전인 2012년에 장애학생들을 대상으로 언플러그드 교육과 텍스트 코딩 교육 연구가 이루어진 것은 고무적인 일이다. 김동일 외(2012)와 김정일 외(2012)의 연구에서는 미래 IT 교육의 중요성 및 필요성을 인식하여 장애학생들에게 SW교육 연구를 실시하였다. SW교육은 고차적 사

고력이 필요한 교육으로 학습자의 학습 수준이 높아야 한다는 인식이 있지만 두 연구를 통해 장애학생들도 SW교육이 가능하고 필요함을 증명하였으며, 이와 더불어 2015 개정 교육과정을 통해 장애학생 대상 SW교육 연구가 적지만 꾸준히 실시되었다.

선행연구의 SW교육 수업유형은 언플러그드 교육, EPL 교육, 피지컬 컴퓨팅 교육, 텍스트 코딩 교육까지 다양하게 연구되어왔으며, 국외 중재 연구에서는 국내 중재 연구와 달리 텐저블 코딩에 관한 연구(Taylor, 2018; Taylor et al., 2017)도 이루어졌다. 텐저블 코딩이란 물리적 조작물을 사용해 책상 위에서 다양한 프로그래밍 요소, 명령, 제어 흐름을 나타낼 수 있는 프로그래밍 언어의 한 종류이다(채성은, 허유성, 2021; Horn & Jacob, 2007). 텐저블 코딩은 텐저블 유저 인터페이스(Tangible User Interface, TUI)기반의 교구를 사용하며 디지털 정보를 물리적 형태로 부여하여 능동적인 조작을 가능하게 한다(권속진, 2021). 책상 위에서 블록 같은 조작물을 이용해 알고리즘을 설계하고, 프로그램 화면을 통해 알고리즘을 확인한다. 언플러그드 교육 방법과 달리 아날로그와 디지털을 융합한 교육방법으로 semi-plugged 교육이라고도 한다. 이러한 장점으로 SW교육을 처음 접하는 장애학생들에게 쉽게 가르칠 수 있을 것으로 예상되어 앞으로 국내에서도 관련 연구가 이루어져야 할 것이다.

선행연구의 연구참여자는 초등학생부터 대학생까지 다양한 학년 군을 대상으로 실시되었으며 초등학생을 대상으로 한 연구가 가장 많았다. 장애 영역은 지적장애가 가장 많았으며, 김동일 외(2012)의 연구에서는 각각 다른 대학교에 재학 중인 시각장애, 청각장애, 지체장애 학생들을 대상으로 연구가 시행되었으며, 이현주 외(2020)의 연구에서는 여러 특수학교에서 지적장애, 시각장애, 청각장애, 지체장애 학생들을 대상으로 연구가 시행되었다.

종속변인은 학업성취도를 측정 한 연구가 가장 많았으며 사후 검사를 통해 학업성취도가 향상되어 SW교육이 장애학생의 SW교육 학업성취도, 알고리즘 학업성취도, 정보처리 학업성취도, 국어 및 수학 학업성취도에 긍정적인 영향을 끼치는 것을 알 수 있었다(김정일 외, 2012; 신성희, 한경근, 2019; 이현주 외, 2020; 정소영, 2019). 학업성취도 외에 여러 연구에서 다양한 종속변인이 측정되었다. 컴퓨팅 사고력, 학습된 무기력, 자발적 언어사용, 사회적 기술, 학습동기, 인지 과정, 코딩 정확도 등이 있었으나 연구의 수는 매우 적어 장애학생 대상 SW교육에 대한 지속적인 연구가 필요함을 시사하였다.

학업성취도 외에 기타 종속변인에 관한 선행연구를 살펴보면 먼저 김동일 외 (2012)는 장애학생들의 고등교육 진학 비율이 높아졌으며, IT 기술이 미래 직업 준비에 유용함을 인식하였고, 앞으로도 장애학생을 위한 IT 교육 프로그램이 필요하므로 고등교육에서 체계적인 IT 교육 프로그램이 필요하다고 보았다. 이 연구에서는 각자 다른 대학교에 재학 중인 시각장애, 청각장애, 지체장애 대학생들을 대상으로 텍스트 코딩 프로그래밍 수업을 하고 면담을 통해 컴퓨터 프로그래밍 수업 경험을 분석하였다. 학생들은 텍스트 코딩 수업 경험을 통해 수업 과정에서 어려움을 경험했으며, 자기주도적 학습방법을 습득하였고, IT 교육에 대한 태도가 긍정적으로 변화되었다고 하였다. 구은정(2016)의 연구에서는 학습장애 초등학생 1명을 대상으로 누리 아띠 로봇을 이용한 피지컬 컴퓨팅 교육을 통해 학습된 무기력과 자발적 언어사용의 변화를 알아보았는데 학습된 무기력이 감소하고 자발적 언어 사용이 증가하였다. 김소연, 민천식(2019)은 지적장애 학생이 포함된 초등학교 6학년 두 학급 중 한 학급에만 언플러그드 교육, EPL 교육, 피지컬 컴퓨팅 교육을 실시하여 통합학급 학생들을 대상으로 장애수용 태도를 연구하였다. 연구 결과 장애학생과 함께 SW교육을 학습한 학급에서 장애학생에 대한 긍정적인 수용 점수가 높게 나왔다. 신성희, 한경근 (2019)은 지적장애 고등학생을 대상으로 언플러그드 교육을 실시하였으며 학업 성취도 뿐만 아니라 사회적 기술의 변화도 함께 연구하였다. 연구 결과 사회적 기술이 사전 검사보다 사후 검사가 높게 나타나 학업성취도 뿐만 아니라 사회적 기술에도 긍정적인 영향을 끼치는 것으로 나타났다. 황규용(2017)은 지적장애 초등학생 1명을 대상으로 피지컬 컴퓨팅 도구인 메이키 메이키를 이용해 시연중심모델(DMM)을 적용하여 피지컬 컴퓨팅 교육을 실시하고 이 때 나타나는 총체적인 경험과 효과를 질적분석을 통해 연구하였다. 연구 결과 의사표현에서는 부정적인 표현이 줄고 긍정적인 표현이 높게 나타났으며, 수업 참여행동에서는 수업 준비, 지시따르기, 과제 수행, 질문에 대답하기 행동에서 유의미한 행동을 보여주었다. 주의집중력 영역에서는 수업자료 관찰, 과제집중, 주의집중시간에서 유의미한 행동을 보여주었다. 홍순재(2019)는 절차적 사고 함양을 위한 모바일 학습 프로그램을 개발하고 지적장애 초등학생 3명에게 프로그램을 적용하여 면담을 통해 연구결과를 도출하였다. 연구 결과 학습 수준과 내용을 조절하면 장애학생도 SW교육 학습이 가능하다고 하였으며, 모바일을 통한 개별화된 교육이 높은 학습몰입과 성취를 가져왔다고 하였다. 또한 SW교육이 교육

과정에서 제시하는 핵심역량과 교과역량을 함양시켰으며 학생들이 주의집중과 관련된 유의미한 행동을 보여주었다고 하였다. 정소영(2019)은 지적장애 초등학생을 대상으로 엔트리 활용 EPL 수업이 학습동기와 학업성취도에 미치는 영향을 연구하였다. 학습동기는 수업태도와 학습만족도를 측정하였는데 수업태도는 집중시간에 변화를 알아보았으며 집중시간이 매 차시마다 향상되었다. 홍다희(2016)는 지적장애 초등학생을 대상으로 언플러그드 교육과 함께 EPL 중 하나인 Code.org를 활용한 SW교육이 인지과정에 미치는 영향을 연구하였다. 인지과정의 변화를 알아보기 위해 종합인지기능 진단검사를 중재 전, 후에 실시하였으며 주의집중과 동시처리 영역에서 2명 학생 모두 사후 검사에서 향상을 보였으며, 계획기능 영역과 순차처리 영역은 각각 1명의 학생이 사후 검사에서 향상을 보였다.

국외 중재 연구를 살펴보면 Taylor(2018)는 다운증후군 초등학생을 대상으로 컴퓨터 프로그래밍 교육을 실시하여 코딩 정확도를 측정하였다. 피지컬 컴퓨팅 교육을 통해 코딩 정확도를 알아보았으며 연구참여자 3명 모두 효과크기가 크게 나타났다. Taylor et al(2017)은 지적장애 초등학생을 대상으로 컴퓨터 프로그래밍 교육을 실시하여 코딩 정확도를 측정하였다. 피지컬 컴퓨팅 교육을 통해 코딩 정확도를 알아보았으며 연구참여자 3명 모두 효과크기가 효과적(1명), 매우 효과적(2명)으로 나타났다. Munoz et al(2018)은 초등학생 및 중학생 자폐성 장애 학생들을 대상으로 게임을 만드는 프로그래밍 수업을 통해 컴퓨팅 사고력의 영향을 연구하였다. 컴퓨팅 사고력 검사 결과 사전 검사 평균 13.9점, 사후 검사 평균 16점으로 사후 검사 점수가 향상되었다. 채성은과 허유성(2021)은 최근 10년간 국내·외 특수교육 분야 SW교육 관련 연구들의 동향을 분석하였다. 그 중 실험연구를 진행한 연구들의 수업유형, 종속변인, 주요 결과는 <표 II-3>과 같다.

장애학생을 대상으로 한 SW교육 선행연구들을 살펴보면 양적인 측면에서 매우 제한적인 것으로 나타났다. 일반교육 분야에서는 컴퓨팅 사고력의 중요성이 대두되고 EPL이 개발되면서 SW교육에 대한 관심이 높아져 많은 연구가 이루어진 반면 특수교사들은 SW교육에 대한 낮은 인식 수준을 보였으며, SW교육이 장애학생 수준에 난이도가 높고 적절하지 않다는 인식이 있었다(김동선, 2020). 하지만 디지털 소양이 기초소양 중 하나로 선정될 만큼 SW교육은 모든 학습자의 보편교육으로서 실행되어야 한다. 따라서 학교 현장에서 장애학생을 위한 S

<표 II-3> 선행연구의 수업유형, 종속변인, 주요 결과

연구자	수업유형	종속변인	주요 결과
구은정 (2016)	피지컬 컴퓨팅	학습된 무기력	• 24점 감소로 긍정적 효과를 보임
		자발적 언어 사용	• 자발적 발화 정도: PND=75%
김동일 외 (2012)	텍스트 코딩	컴퓨터 프로그래밍 수업 경험	<ul style="list-style-type: none"> • 수업 과정에서의 어려움: 내용 이해 어려움, 실습시간 부족, 장애에 대한 강사 인식 부족, 장애 관련 지원 부족 • 습득도움요인: 자기주도적 학습방법 • 수업성과: IT 교육 내용에 대한 긍정적 태도 변화, 진로에 대한 비전 설정, IT 관련 지식 습득
김소연, 민천식 (2019)	<ul style="list-style-type: none"> • 언플러그드 • 피지컬컴퓨팅 • EPL 	장애수용 태도	<ul style="list-style-type: none"> • 통합에 대한 태도: 실험집단 효과적 ($t = 2.0, p > .05$) • 장애학생의 능력에 대한 인식: 실험집단 효과적 ($t = 2.6, p > .05$) • 장애학생에 대한 행동의도: 실험집단 효과적 ($t = 2.1, p > .05$) • 학급동료 인식 기대: 실험집단 효과적 ($t = 2.3, p > .05$)
신성희, 한경근 (2019)	언플러그드	알고리즘 학업 성취도	• 사후 검사결과 평균 60점 향상함
		사회적 기술	SRSS <ul style="list-style-type: none"> • 자기주장: 사후 점수 8.3점 향상 • 협력: 사후 점수 10.3점 향상 • 자아통제: 사후 점수 5.7점 향상 사회적 기술 행동관찰 <ul style="list-style-type: none"> • 자기주장: 사후 점수 10.7점 향상 • 협력: 사후 점수 10.4점 향상 • 자아통제: 사후 점수 6.4점 향상
김정일 외 (2012)	언플러그드	학업성취도	• 실험집단이 높게 나타남($t = 2.1, p > .05$)

<표 II-3> 선행연구의 수업유형, 종속변인, 주요 결과 (계속)

연구자	수업유형	종속변인	주요 결과
이현주 외 (2020)	<ul style="list-style-type: none"> 언플러그드 EPL 텍스트 코딩 피지컬컴퓨팅 	학업성취도	<ul style="list-style-type: none"> 지적장애학생: 82.3%가 ‘중’ 수준 이상 성취도 달성 청각장애학생: 90.9%가 ‘중’ 수준 이상 성취도 달성 지체장애학생: 78.8%가 ‘중’ 수준 이상 성취도 달성 시각장애학생: 78.8%가 ‘중’ 수준 이상 성취도 달성
황규용 (2017)	피지컬 컴퓨팅	총체적 경험	<ul style="list-style-type: none"> 의사표현: 부정적 표현이 줄고, 긍정적 표현이 늘어남 수업 참여행동: 수업준비, 지시따르기, 과제 수행, 질문에 대답 등 유의미한 행동 보여줌 주의집중: 수업자료 관찰, 과제 집중, 주의집중 시간 등 유의미한 행동 보여줌
홍순재 (2019)	모바일 학습	절차적 사고	<ul style="list-style-type: none"> 장애학생도 소프트웨어 교육 학습이 가능함 모바일 통한 개별화교육이 높은 학습몰입과 성취를 가져옴 주의집중 관련 유의미한 행동을 보여줌
정소영 (2019)	EPL	학업성취도	<ul style="list-style-type: none"> 국어: 사후 점수 평균 21.4점 향상 수학: 사후 점수 평균 21.6점 향상 프로그래밍 검사: 평균 82점
		학습동기	<ul style="list-style-type: none"> 집중시간: 매차시마다 향상 수업에 대한 긍정적 답변: 학생 90%, 교사 96%
홍다희 (2016)	<ul style="list-style-type: none"> 언플러그드 EPL 	인지과정	<ul style="list-style-type: none"> 계획기능: 아동A 사후 검사 향상 주의집중: 아동A, B 사후 검사 향상 동시처리: 아동A, B 사후 검사 향상 순차처리: 아동B 사후 검사 향상
Taylor (2018)	<ul style="list-style-type: none"> 피지컬 컴퓨팅 텐저블 코딩 	코딩정확도	<ul style="list-style-type: none"> P1아동: PND=100%, 효과크기=0.86 P2아동: PND=100%, 효과크기=0.92 P3아동: PND=100%, 효과크기=1.0
Taylor et al.(2017)	<ul style="list-style-type: none"> 피지컬 컴퓨팅 텐저블 코딩 	코딩정확도	<ul style="list-style-type: none"> P1아동: PND=100%, 효과크기=0.95 P2아동: PND=100%, 효과크기=1.00 P3아동: PND= 84%, 효과크기=1.00
Munoz et al.(2018)	블록 코딩	컴퓨팅 사고력	<ul style="list-style-type: none"> 사후 검사 점수 평균 2.1점 향상

출처: 채성은, 허유성(2021)의 수업유형별 종속변인 및 주요결과 재구성

W교육이 자리 잡기 위해서는 특수교사의 SW교육에 대한 관심과 교수 역량을 높이는 방안이 필요하다. 또한 교원양성기관에서는 예비 특수교사를 대상으로 SW교육에 대한 강의를 제공하여 학교 현장에서 SW교육을 교과교육에 융합하여 장애학생들의 학업 능력을 증진시킬 수 있는 교사로 양성해야 할 것이다.

선행연구의 SW교육 종류를 보면, 언플러그드 교육만을 활용한 중재 연구가 2편(김정일 외, 2012; 신성희, 한경근, 2019), EPL 교육만을 적용한 중재 연구가 1편(정소영, 2019), 그 외에 언플러그드 교육과 EPL 교육을 함께 적용한 연구가 3편(김소연, 민천식, 2019; 이현주 외, 2020; 홍다희, 2016)으로 매우 적은 연구가 이루어졌다. 또한 두 중재 간에 장애학생에게 미치는 상대적 교육 효과를 비교하거나 장애학생의 수업 참여 양상(예: 수업 참여행동 등) 등을 체계적으로 비교 분석한 연구는 없었다.

한편 선행연구들에 따르면 장애학생을 대상으로 실시한 SW교육은 다양한 중속변인에서 긍정적인 효과를 나타냈다. SW교육은 컴퓨팅 사고력을 신장시키며 알고리즘 사고에 긍정적인 영향을 미치고(신성희, 한경근, 2019; Munoz et al., 2018), 학습된 무기력이 줄어들며(구은정, 2016), 학업 태도와 사회적 기술 등이 향상되는 긍정적인 결과를 보여주었다(신성희, 한경근, 2019; 정소영, 2019). 중속변인을 살펴보면 컴퓨팅 사고력, 주의집중, 수업 참여행동에 관한 연구가 있었지만, 문제해결력에 관한 연구는 없었다. 이 연구는 SW교육의 수업 유형 중 하나인 언플러그드 교육과 EPL 교육을 지적장애 초등학생에게 실시하여 두가지 교육방법이 문제해결력과 수업 참여행동에서 어떤 차이를 나타내는 지 알아보고자 하였으며, 컴퓨팅 사고력에 효과가 있는지 살펴보고자 한다.

III. 연구 방법

1. 연구 설계

이 연구는 장애학생 3명을 대상으로 단일대상설계와 사전-사후 검사를 적용하였다. 이 연구의 독립변인은 언플러그드 교육과 EPL 교육이며, 종속변인은 문제해결력, 수업 참여행동, 컴퓨팅 사고력이다.

먼저 언플러그드 교육과 EPL 교육이 문제해결력과 수업 참여행동에 미치는 효과를 비교하기 위해 교대중재설계(alternating treatments design)를 사용하였다. 교대중재설계는 한 가지 종속변인에 대한 하나 이상의 중재 전략의 효과를 비교하는 단일대상 실험설계이며 기초선 측정을 반드시 하지 않아도 되며, 시험 종료 시기에 따른 타당도 문제가 다른 설계에 비해 적다는 장점이 있다(이소현 외, 2000). 따라서 컴퓨팅 기기를 사용하지 않는 언플러그드 교육과 컴퓨팅 기기를 사용하는 EPL 교육 중 어느 교육 방법이 장애학생에게 효과적인지 알고자 하는 이 연구에 부합한다.

실험은 기초선, 중재, 효과적인 처치의 순서로 실행하였다. 언플러그드 교육과 EPL 교육의 순서는 임의적 순서로 배치하여 각 중재가 다른 중재에 영향을 끼치는 것을 최소화시켰다. 그리고 기능적 관계 입증을 위해 중재가 끝난 후 효과적인 처치 구간을 도입하였다(Alberto & Troutman, 2010).

문제해결력의 경우 기초선과 효과적인 처치 구간을 도입하지 않고 중재 회기만을 측정하여 두 중재를 비교하였다. 문제해결력 측정 문항은 알고리즘을 설계하는 문제로 SW교육을 실시한 후 풀 수 있는 문제이다. 따라서 컴퓨팅 사고력의 사전-사후 검사에 영향을 끼칠 수 있으므로 기초선을 도입하지 않았다. 또한 EPL 교육의 알고리즘 문제해결 활동에서 프로그래밍 문제해결 활동이 난이도 순으로 진행되기 때문에 알고리즘 문제 활동이 EPL 8차시로 완료되어 효과적인 처치 구간에서는 알고리즘 문제해결력을 측정하지 않았다. 수업 참여행동의 경우 기초선, 중재, 효과적인 처치 구간에서 자발적 질문, 주의집중 행동, 과제회피 행동을 측정하였다.

다음으로 언플러그드 교육과 EPL 교육, 즉 SW교육이 컴퓨팅 사고력에 영향을 미치는지 알아보기 위해 사전-사후검사를 실시하였다. 컴퓨팅 사고력은 단일한

하나의 역량으로 측정되는 것이 아니며 컴퓨팅 사고력을 구성하는 하위 역량들을 측정하여 그 효과성을 검증할 수 있다(김민정 외, 2017). 언플러그드 교육과 EPL 교육은 매 차시에 모든 컴퓨팅 요소를 학습하는 것이 아니라 주요 요소 1~2가지씩을 선정하여 교육하기 때문에 모든 차시가 끝나야 컴퓨팅 사고력의 주요 요소를 학습했다고 볼 수 있다. 언플러그드 교육 혹은 EPL 교육이 끝나는 시점이 컴퓨팅 사고력의 주요 요소들이 전부 학습된 상태이므로 매 차시 검사를 실시하여 그래프로 나타내는 방법은 부적합하다. 따라서 언플러그드 교육과 EPL 교육을 실시하기 전-후를 비교하여 컴퓨팅 사고력의 변화가 나타나는지 알아보고자 하였다. 이 연구의 종속변인별 실험설계 및 분석방법은 <표 III-1>과 같다.

<표 III-1> 종속변인별 실험설계 및 분석방법

종속변인	실험설계	분석방법
문제해결력	교대중재	중재 회기만 비교
수업 참여행동	교대중재	기초선, 중재, 효과적인 처치 비교
컴퓨팅 사고력	사전-사후	사전-사후 검사 결과 비교

2. 연구참여자

이 연구의 참여자는 A광역시에 소재하는 초등학교에 재학 중인 지적장애 학생 3명이다. 구체적인 연구참여자 선정기준은 다음과 같다.

첫째, 「장애인 등에 대한 특수교육법」에 의거하여 특수교육대상자로 선정되고 특수학급에 배치된 지적장애 초등학생이다.

둘째, 문장을 읽고 의미 파악이 70% 정도 가능하며, 2022학년도 IEP(개별화 교육계획)에 준하여 국어는 1학년 2학기~2학년 1학기, 수학은 2학년 2학기~3학년 1학기의 진도를 나가는 학습 수준이 비슷한 학생이다. 왜냐하면, EPL 블록이 글자로 제시되어 있어 각 블록의 의미를 파악할 수 있어야 하며, 언플러그드 반복 명령어 카드가 X2, X3으로 표시되기 때문이다.

셋째, 컴퓨터, 스마트폰, 태블릿 PC 사용 경험이 있는 학생이다. 왜냐하면, EPL 교육이 인터넷 플랫폼을 활용해야 하므로 기본적인 컴퓨팅 기기의 사용능

력이 필요하기 때문이다.

넷째, 다른 감각장애나 신체장애가 없는 학생이다. 이 연구의 EPL 교육에서 사용할 컴퓨팅 기기는 태블릿 PC로 선정하였으므로 태블릿 PC 화면 터치 및 조작에 어려움이 없어야 하기 때문이다.

다섯째, SW교육 경험이 없는 학생이다.

여섯째, 방과 후에 실험연구 수업에 참여 가능한 학생이다.

일곱째, 본 실험연구의 참여에 보호자의 동의가 이루어진 학생이다

연구참여자의 기본 정보 및 특성은 <표 III-2>와 같다.

<표 III-2> 연구참여자 기본 정보 및 특성

학생별 특성	학생A	학생B	학생C
연령(성별)	만 10세 (남)	만 11세 (남)	만 11세 (남)
학년	초등학교 4학년	초등학교 5학년	초등학교 5학년
보건복지부 장애등록명 및 정도	장애등록 하지 않음	지적장애 3급	지적장애 3급
장애인 등에 대한 특수교육법 장애명	지적장애	지적장애	지적장애
특수학급	K-WPPSI-IV	K-WPPSI-IV	K-WISC-III
입급 당시	전체 IQ 74	전체 IQ 54	전체 IQ 57
검사 결과	KNISE-SAB	KNISE-SAB	SMS
	전체적응행동 90	전체적응행동 53	SQ 57
문제해결력	<ul style="list-style-type: none"> 전 차시에서 배운 내용을 상기하여 문제해결 전략을 세울 수 있음. 문제에서 구하려는 것이나 조건을 확인하지 않으며 문제해결의 신속성에만 신경 씀. 문제해결 과정의 검토를 요구하면 검토 행동을 가장(假裝)하거나 과제회피 행동이 나타남. 	<ul style="list-style-type: none"> 문제에서 구하려는 것이나 조건을 세심하게 확인하지 않음. 스스로 문제해결 전략을 세우기보다 타인의 도움을 요청하여 해결하려는 의존적인 성향임. 검토과정의 필요성을 느끼지 못하고 수행하지 않음. 	<ul style="list-style-type: none"> 전 차시에서 배운 내용을 기억하여 문제해결 전략을 세울 수 있으나, 문제해결의 정확성보다는 신속성에 초점을 맞추는 경향이 있음. 오답임을 알았을 때 큰 좌절감을 느끼며, 인내심이 약하여 문제해결 과정을 검토하는 것을 거부함.

<표 III-2> 연구참여자 기본 정보 및 특성 (계속)

학생별 특성	학생A	학생B	학생C
자발적 질문	<ul style="list-style-type: none"> 수업 상황에서 자신이 알고 있는 사실과 다른 것을 찾아 질문할 때가 가끔 있음. 외부자극에 둔감한 편이며, 교실의 배치나 또래의 외모 등과 같은 변화를 감지해 질문할 때가 가끔 있음. 	<ul style="list-style-type: none"> 수업 상황에서 질문보다는 ‘몰라요’로 표현함. 타인에 관한 관심을 가지고 질문하거나 어떠한 현상의 궁금증을 가지고 질문하는 경우가 간혹 있음. 교실의 배치, 또래의 외모 등의 변화를 감지해 질문할 때가 가끔 있음. 	<ul style="list-style-type: none"> 과묵한 성격으로 말수가 적어 수업 상황에서 학습에 대해 질문을 하는 경우가 거의 없으며 교사가 먼저 질문할 때까지 기다림. 타인에 관한 관심을 가지고 질문하거나 어떠한 현상의 궁금증을 가지고 질문하는 경우는 없음.
주의집중 행동	<ul style="list-style-type: none"> 바르게 착석하는 시간이 짧고, 주의집중 유지시간이 매우 짧음. 교사의 관심이 없을 때 과제회피 행동을 하기 위해 과제보다는 교사에게 주의집중하는 시간이 많음. 	<ul style="list-style-type: none"> 바르게 착석할 수 있나 자신의 과제에만 집중하는 선택적 주의집중능력이 약하며, 주의집중 시간이 길어지면 과제를 포기함. 	<ul style="list-style-type: none"> 선호하는 활동에서는 주의집중 유지시간이 길게 나타남. 청각적 주의력이 약하여 듣기평가가 약하고 지시사항을 잘 이행하지 못하며 불필요한 청각자극에 주의를 기울임.
과제회피 행동	<ul style="list-style-type: none"> 연필과 지우개로 서로 싸우는 동작을 반복하거나, 손을 이용해 특정한 동작을 반복함. 	<ul style="list-style-type: none"> 앞아서 몸을 앞뒤, 좌우로 흔들거나, 옆자리 학생의 과제 수행을 참견하고 행동을 지적함. 	<ul style="list-style-type: none"> 아무 행동도 하지 않고 멍하니 앉아 있음.
컴퓨팅 사고력	<ul style="list-style-type: none"> 컴퓨팅 사고력 사전 검사 결과 패턴인식, 모델링 및 시뮬레이션 영역에 대한 문제를 풀지 못함. 	<ul style="list-style-type: none"> 컴퓨팅 사고력 사전 검사 결과 패턴인식에 대한 문제를 풀지 못함. 	<ul style="list-style-type: none"> 컴퓨팅 사고력 사전 검사 결과 패턴인식에 대한 문제를 풀지 못함.

가. 학생A

학생A는 장애 등록은 하지 않았지만 초등학교 2학년 때 특수교육대상자로 의뢰되어 지적장애로 선정되었다. 2학년까지는 특수교육 관련서비스 지원만 받았으나 한글을 쓰고 읽는데 지속적인 어려움을 보여 초등학교 3학년부터 특수교육을 받고 있다. 학생A는 「국어」, 「수학」 교과에서 전 차시에서 배운 내용을 상기하여 문제해결 전략을 세울 수 있으나 문제 풀이의 정확성에는 관심이 없고 신속성에만 집중한다. 교사가 문제해결 후 검토를 요구하지만 검토 행동을 가장할 뿐 실제로 수행하지는 않는다. 수업 상황에서 자신이 알고 있는 사실과 다를 경우 질문할 때가 간혹 있으나 그 빈도는 매우 적다. 외부자극에 둔감한 편이지만 변화를 감지하여 질문할 때가 간혹 있다. 주의집중 시간이 매우 짧고 과제 회피행동을 하기 위해 교사의 눈치를 보느라 과제보다는 교사에게 집중하는 시간이 길다. 과제회피 행동으로는 책상에 조용히 앉아 할 수 있는 행동들을 한다. 연필과 지우개로 싸우는 동작을 반복하거나, 자신이 만든 손 동작을 끊임없이 반복한다. 4학년 학생으로 아직 SW교육 경험이 없으며 컴퓨팅 사고력 사전 검사 결과 패턴인식, 모델링 및 시뮬레이션 영역의 문제를 풀지 못하는 모습을 보였다.

나. 학생B

학생B는 지적장애로 장애등록을 한 학생이며 초등학교 1학년부터 특수교육대상자로 의뢰되었고 특수학급에 배치되어 특수교육을 받고 있다. 학생B는 「국어」, 「수학」 교과에서 문제에서 요구하는 조건을 세심하게 확인하지 않으며 문제를 보는 즉시 교사에게 도움을 구하는 의존성을 보인다. 교사가 문제해결 후 검토를 요구하지만, 인내심이 약하여 검토를 수행하지 않는다. 수업 상황에서 ‘몰라요’ 라는 말을 자주 사용하며 질문하는 행동의 빈도는 매우 적다. 외부자극에 둔감하여 변화를 감지 질문하는 행동은 극히 드물다. 과제에 집중하는 시간이 짧고 주의집중 시간이 길어질 때 과제를 포기한다. 과제회피 행동으로는 상체를 앞-뒤, 혹은 좌-우로 반복해서 흔드는 모습을 보이며, 옆자리 학생의 행동이나 과제를 참견하고 지적한다. 5학년 학생이지만 재학 중인 학교에

서는 6학년부터 SW교육을 실시하므로 SW교육 경험이 없으며 컴퓨팅 사고력 사전 검사 결과 패턴인식에 대한 문제를 풀지 못하였다.

다. 학생C

학생B는 지적장애로 장애등록을 한 학생이며 초등학교 1학년부터 특수교육대상자로 의뢰되었고 특수학급에 배치되어 특수교육을 받고 있다. 과묵한 성격으로 먼저 말을 꺼내는 때는 없으며 그로 인해 스스로 질문하는 때도 없고, 교사의 질문에 대답하지 않을 때가 많다. 선호하는 활동에서는 주의집중 유지 시간이 길며 선호하지 않는 「국어」 교과 시간에는 주의집중 시간이 짧다. 청각적 주의력이 약하여 불필요한 자극에 주의를 집중하고 교사의 지시를 이행하지 않을 때가 많다. 과제회피 행동으로는 아무 행동도 하지 않고 가만히 앉아 있는 모습을 보인다. 5학년 학생이지만 재학 중인 학교에서는 6학년부터 SW교육을 실시하므로 SW교육 경험이 없으며 컴퓨팅 사고력 사전 검사 결과 패턴인식에 대한 문제를 풀지 못하였다.

3. 독립 변인

가. 중재 절차

이 연구의 목적은 교사들에게 잘 알려진 SW교육 방법 중 지적장애 초등학생을 대상으로 언플러그드 교육과 EPL 교육을 실시하여 두 중재 간의 상대적 효과를 비교하는 것이다. 중재 프로그램 개발을 위해 지도 교수 1인과 B교육대학교 컴퓨터교육과를 졸업하였고 5~6학년 「실과」 교과의 SW교육을 가르친 경험이 있는 교육경력 6년차인 초등교사 1인에게 자문을 구하여 진행하였다. 중재 프로그램은 장애학생을 대상으로 제작된 검증된 SW교육 프로그램을 선정하고자 하였다. 따라서 장애학생의 교육기회 확대를 위해 장애학생 교수·학습을 지원하는 국립특수교육원의 SW교육 프로그램을 중재 프로그램으로 선정하였다. 국립특수교육원 홈페이지에서 배포한 SW교육 콘텐츠 및 교재인 ‘나랑 놀자! 소프트웨어 입문편(발달장애학생용)’ 과 국립특수교육원(2019)에서 장애학생 교

<표 III-3> 언플러그드 교육과 EPL 교육의 회기별 교육 활동

구분	회기	유형	교육 활동	측정
기초선	1	해당 없음	• 소프트웨어의 의미와 생활 속 소프트웨어 알기	• 수업 참여행동
	2		• 컴퓨터의 입력장치와 출력장치 알기	
	3		• 태블릿 PC 사용방법 알기	
중재	1	언플러그드	• 패턴을 반복해 사탕 가랜드 만들기 • 알고리즘(순차) 문제 해결하기	• 문제해결력 • 수업 참여행동
	2	EPL	• 엔트리 활용 방법 알기 • 알고리즘(순차) 문제 해결하기	
	3	EPL	• 동물 친구 이야기 만들기 • 알고리즘(순차) 문제 해결하기	
	4	언플러그드	• kidbot 문제 해결하기 • 알고리즘(순차) 문제 해결하기	
	5	EPL	• 코딩으로 만드는 자기소개 • 알고리즘(순차) 문제 해결하기	
	6	EPL	• 색이 변하는 변신 로봇 만들기 • 알고리즘(순차) 문제 해결하기	
	7	언플러그드	• kidbot 문제 해결하기 • 알고리즘(순차) 문제 해결하기	
	8	언플러그드	• kidbot 문제 해결하기 • 알고리즘(순차) 문제 해결하기	
	9	EPL	• 순서대로 우주 여행하기 • 알고리즘(순차, 반복) 문제 해결하기	
	10	언플러그드	• kidbot 문제 해결하기 • 알고리즘(순차, 반복) 문제 해결하기	
	11	언플러그드	• 절차에 맞춰 화분설명서 만들기 • 알고리즘(순차, 반복) 문제 해결하기	
	12	EPL	• 파리잡기 놀이 만들기 • 알고리즘(순차, 반복) 문제 해결하기	
	13	EPL	• 바닷 속 친구들 모습 만들기 • 알고리즘(순차, 반복) 문제 해결하기	
	14	언플러그드	• 정렬망을 이용해 키 순서대로 나열하기 • 알고리즘(순차, 반복) 문제 해결하기	
	15	EPL	• 동물 달리기 만들기 • 알고리즘(순차, 반복) 문제 해결하기	
	16	언플러그드	• 점 잇기로 그림 완성하기 • 알고리즘(순차, 반복) 문제 해결하기	
효과적인 처치	1	EPL	• 조건을 생각하기	• 수업 참여행동
	2		• 신비한 블록과 놀기	
	3		• 똑똑한 블록과 놀기	

과연계 교수·학습 보완자료로 발간한 ‘소프트웨어(SW)와 친구하자’가 1차로 선정되었고, 2차 회의를 통해 기초선, 중재, 효과적인 처치에 들어갈 차시를 선정하였다. 연구참여자 선정 후 3차 회의를 통해 추출한 차시를 연구참여자의 학습 수준과 장애 특성을 고려하여 프로그램 수정·보완 작업을 시행하였다. 이 연구의 언플러그드 교육과 EPL 교육의 회기별 활동은 <표 III-3>과 같다.

1) 기초선

기초선에서는 소프트웨어에 관한 기초 소양 교육 및 EPL 교육 연구 도구인 태블릿 PC 사용법을 교육하였다. 이 연구의 연구참여자는 스마트폰 사용 경험은 많았고, 컴퓨터 및 태블릿 PC는 학교 수업 중 사용해 본 경험이 1년에 2~3번 정도 있었다. 이 연구의 EPL 교육은 태블릿 PC를 이용하기 때문에 태블릿 PC 사용법을 기초선 3차시에 지도하였다. 기초선 기간의 교육내용은 국립특수교육원(2019)에서 장애학생 교과연계 교수·학습 보완자료로 발간한 ‘소프트웨어(SW)와 친구하자’에서 관련 내용 3차시를 선정하여 연구참여자의 장애 정도와 학습 수준에 맞게 수정·보완하였다. 기초선 기간에는 수업 참여행동을 측정하였으며, 기초선 기간이 끝난 후 중재에 들어가기 전 컴퓨팅 사고력 사전 검사를 실시하였다. 기초선 구간에서 실시한 교육내용은 <표 III-4>와 같다.

<표 III-4> 기초선 기간 교육내용

차시	교육내용	사용 기기
1차시	소프트웨어의 의미와 생활 속 소프트웨어 알기	
2차시	컴퓨터의 입력장치와 출력장치 알기	컴퓨터
3차시	태블릿 PC 사용방법 알기	태블릿 PC

2) 중재

이 연구에서는 언플러그드 교육과 EPL 교육의 효과를 비교하는 연구이므로 다음과 같은 차이를 중점으로 프로그램을 개발하였다. 첫째, 학습 도구에 차이를 두었다. 언플러그드 교육은 게임이나 놀이를 통해 배우는 방법이므로 활동

지, 화분 만들기 키트, 낚시 놀이 도구, 색연필 등의 도구를 선정하였으며, EPL 교육에서는 특정 EPL 플랫폼에서만 학습이 이루어지므로 EPL 사이트는 ‘나랑 놀자! 소프트웨어 입문편(발달장애학생용)’에서 제시된 엔트리로 선정하였고, 컴퓨팅 기기는 스마트폰 사용에 익숙한 연구참여자의 특성을 고려하여 태블릿 PC로 선정하였다. 둘째, 교육 내용에 차이를 두었다. 각 수업유형의 특성에 따라 언플러그드 교육은 컴퓨터 과학의 원리와 개념 및 알고리즘 학습, EPL 교육은 프로그래밍 교육 및 알고리즘 학습으로 선정하였다. 셋째, 교육 방식에 차이를 두었다. 각 수업유형의 특성에 따라 언플러그드 교육은 놀이, 만들기 등의 조작활동 등의 컴퓨터 과학의 원리와 개념 학습, 알고리즘 해결안 발표로 구성하였고, EPL 교육은 프로그래밍 교육과 엔트리 사이트에서 제시한 문제의 알고리즘을 설계하고 알고리즘 해결안을 발표하는 것으로 구성하였다. 두 교육 방법에 대한 차이를 요약한 표는 <표 III-5>와 같다.

<표 III-5> 두 교육 방법의 차이

영역	언플러그드 교육	EPL 교육
학습 도구	활동지, 화분 만들기 키트, 사탕 가랜드 만들기 키트, 낚시대, 색연필 등	<ul style="list-style-type: none"> • 기기: 태블릿 PC • 교육플랫폼: 엔트리
교육 내용	<ul style="list-style-type: none"> • 컴퓨터 과학의 원리와 개념 • 알고리즘(순차, 반복) 	<ul style="list-style-type: none"> • 프로그래밍 교육 • 알고리즘(순차, 반복)
교육 방식	놀이, 조작 활동, 문제해결 및 발표	문제해결 및 발표

(가) 언플러그드 교육 중재

언플러그드 교육이란 컴퓨터 과학의 기본 원리, 알고리즘적 사고, 프로그래밍 원리를 컴퓨터나 전문 장비 없이 게임이나 놀이를 통해 배우는 방법이다. 이 연구의 언플러그드 교육은 CS Unplugged 웹사이트(csunplugged.org)의 알고리즘을 학습하는 키드봇 영역과 2019년 국립특수교육원, Microsoft, JA Korea가 공동개발한 ‘나랑 놀자! 소프트웨어 입문편(발달장애학생용)’의 콘텐츠 중 1단원 ‘재미있는 언플러그드’ 중 반복 나열, 패턴 이해를 학습하는 3차시, 정렬에 대해 학습하는 7차시, 순차에 대해 학습하는 8차시, 규칙과 절차적 사고를 학습하

는 12차시를 선정하였다. 연구참여자의 장애 정도와 학습 수준에 맞게 수정·보완하였고 교구의 난이도를 재구성하였다. 이 연구의 언플러그드 교육 활동내용은 <표 III-6>, 언플러그드 교육 예시 지도안은 <표 III-7>과 같다.

<표 III-6> 언플러그드 교육 활동내용

차시	활동명	활동내용	컴퓨팅 사고력 요소
1	사탕 가랜드 만들기	<ul style="list-style-type: none"> 반복되는 덩어리를 찾아 규칙 말하기 사탕을 규칙적으로 나열하고 묶으며 사탕 가랜드 만들기 순차구조 알고리즘으로 ‘롤러코스터 타기’ 미션1 해결하기 	반복 나열, 패턴 이해
2	kidbot 문제해결하기 (1)	<ul style="list-style-type: none"> 순차 명령어 카드 개념 이해하기 kidbot 문제해결하기(순차구조) 순차구조 알고리즘으로 ‘누리호 발사’ 미션2 해결하기 	알고리즘(순차)
3	kidbot 문제해결하기 (2)	<ul style="list-style-type: none"> kidbot 문제해결하기(순차구조) 순차구조 알고리즘으로 ‘롤러코스터 타기’ 미션3 해결하기 	알고리즘(순차)
4	kidbot 문제해결하기 (3)	<ul style="list-style-type: none"> 반복 명령어 카드 개념 이해하기 kidbot 문제해결하기(순차, 반복구조) 순차구조 알고리즘으로 ‘누리호 발사’ 미션4 해결하기 	알고리즘(순차, 반복)
5	kidbot 문제해결하기 (4)	<ul style="list-style-type: none"> kidbot 문제해결하기(순차, 반복구조) 순차구조 혹은 반복구조 알고리즘으로 ‘롤러코스터 타기’ 미션5 해결하기 	알고리즘(순차, 반복)
6	화분 설명서 만들기	<ul style="list-style-type: none"> 화분 만드는 순서 알아보고 설명서 작성하기 친구가 작성한 화분 만들기 설명서 보고 순서에 맞춰 따라 만들기 순차구조 혹은 반복구조 알고리즘으로 ‘누리호 발사’ 미션6 해결하기 	규칙, 절차적 사고
7	물고기 키 재기	<ul style="list-style-type: none"> 낙시를 하여 여러 가지 물고기 잡고 길이 순서대로 나열하기 정렬망을 이용해 길이가 다른 빨대를 키 순서대로 정렬하기 순차구조 혹은 반복구조 알고리즘으로 ‘롤러코스터 타기’ 미션7 해결하기 	정렬
8	공룡을 그려요	<ul style="list-style-type: none"> 순서대로 점을 이어 여러 가지 그림 완성하고 발표하기 순차구조 혹은 반복구조 알고리즘으로 ‘누리호 발사’ 미션8 해결하기 	알고리즘(순차)

<표 III-7> 언플러그드 교육 예시 지도안

주제		절차적 사고 이해	회기 (시간)	언플러그드 6회기 (40분)
학습목표		화분 설명서를 만들고 설명서 순서대로 화분을 만들 수 있다.		
컴퓨팅 사고력		규칙, 절차적 사고		
단계 (시간)		교수-학습활동		자료 및 유의점
활동 과정 및 내용	도입 (5)	<ul style="list-style-type: none"> ● 아빠로봇이 왜 샌드위치를 만들지 못했는지 알아보기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"> <p style="text-align: center;">설명서를 보고 화분을 순서대로 만들어보자</p> </div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		☑동영상
	전개 (30)	<p><활동1> 화분 설명서 만들기</p> <ul style="list-style-type: none"> - 교사의 화분 만들기 설명서 제시하고 설명서 만드는 방법에 대해 알기 - 자신이 만들고 싶은 화분을 고르고 화분 설명서 만들기 - 자신의 화분 만들기 설명서 발표하기 <p><활동2> 순서대로 만들기</p> <ul style="list-style-type: none"> - 준비뽑기로 친구의 설명서 뽑기 - 친구의 설명서를 보고 순서대로 화분 만들기 - 자신이 만든 화분 발표하기 - 친구가 자신의 설명서대로 화분을 만들었는지 평가하기 <p><활동3> 누리호 발사 미션 해결하기</p> <ul style="list-style-type: none"> - 누리호 발사 미션6 해결하고 발표하기 		☑ 선생님의 화분 만들기 설명서, 화분 만들기 설명서 활동지 ☑ 화분 만들기 설명서, 화분 만들기 키트 ☑ 알고리즘판, 명령어카드, 문제ppt
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 설명서 만들기와 실제로 만들 때의 주의점 발표하기 		
유의점		<ul style="list-style-type: none"> • 자신이 구상한 화분의 모습과 친구가 만든 화분의 모습이 같지 않을 때 설명서를 보며 이유를 찾게 하고, 설명서를 자세하게 써야 함을 알 수 있게 하여 명령어의 중요성을 알게 한다. • 학습 정리에서 설명서를 순서대로 쓰지 않거나, 만들 때 순서를 따르지 않으면 원하는 작품이 나오지 않음을 알 수 있게 한다. 		

(나) EPL 교육 중재

EPL 교육은 알고리즘적 사고 및 프로그래밍 교육을 교육용 프로그래밍 언어로 배우는 것을 말한다. 이 연구의 EPL 교육은 국립특수교육원(2019)에서 장애 학생 교과연계 교수·학습 보완자료로 발간한 ‘소프트웨어(SW)와 친구하자’의 2장 놀이를 통해 문제 해결하기의 1단원과 2단원, 2019년 국립특수교육원, Microsoft, JA Korea가 공동개발한 ‘나랑 놀자! 소프트웨어 입문편(발달장애 학생용)’의 콘텐츠 중 5단원의 1-3, 1-5 ~ 1-7 차시, 엔트리 사이트의 엔트리 학습하기 첫걸음에 제시된 ‘이상한 숲 속의 엔트리봇’ 미션 8문제를 선정하였고, 학생의 장애 정도와 학습 수준에 맞게 재구성하였다. 이 연구의 EPL 교육 활동내용은 <표 III-8>, EPL 교육 예시 지도안은 <표 III-9>와 같다.

<표 III-8> EPL 교육 활동내용

차시	활동명	활동내용	컴퓨팅 사고력 요소
1	엔트리와 친해지기	<ul style="list-style-type: none"> 엔트리 주요 화면의 요소 이름과 내용 알기 블록을 조립하고 실행화면에서 엔트리봇 움직임 확인하기 순차구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션1 해결 및 발표하기 	EPL 기초개념, 절차적 사고, 알고리즘(순차)
2	동물 친구 이야기 만들기	<ul style="list-style-type: none"> 강아지와 고양이의 말과 대화 순서를 정하고 순차 구조를 이용해 대화 주고 받는 장면 만들기 순차구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션2 해결 및 발표하기 	절차적 사고, 알고리즘(순차)
3	코딩으로 만드는 자기소개	<ul style="list-style-type: none"> 나를 나타내는 오브젝트와 배경화면 선택하고, 순차구조를 이용해 자기소개하기 순차구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션3 해결 및 발표하기 	절차적 사고, 알고리즘(순차)
4	변신 로봇 만들기	<ul style="list-style-type: none"> 순차구조를 이용해 로봇 오브젝트의 색과 크기 바꾸기 순차구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션4 해결 및 발표하기 	절차적 사고, 알고리즘(순차)

<표 III-8> EPL 교육 활동내용 (계속)

차시	활동명	활동내용	컴퓨팅 사고력 요소
5	우주여행 하기	<ul style="list-style-type: none"> • 행성 이동 순서의 순차를 확인하고 명령 순서 정해 우주선 이동시키기 • 순차구조 혹은 반복구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션5 해결 및 발표하기 	반복 나열, 패턴 이해, 알고리즘 (순차, 반복)
6	파리 잡기 놀이 만들기	<ul style="list-style-type: none"> • ‘반복’ 블록, ‘화면 끝에 닿으면 튕기기’ 블록을 사용해 파리 잡기 놀이 만들기 • 순차구조 혹은 반복구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션6 해결 및 발표하기 	반복 나열, 패턴 이해, 알고리즘 (순차, 반복)
7	바닷속 친구들 모습 만들기	<ul style="list-style-type: none"> • 바닷속 물고기들이 자유롭게 이동하는 움직임을 반복 블록으로 표현하기 • 순차구조 혹은 반복구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션7 해결 및 발표하기 	반복 나열, 패턴 이해, 알고리즘 (순차, 반복)
8	동물 달리기 만들기	<ul style="list-style-type: none"> • 동물의 속도를 다르게 하여 두 동물의 달리기 시합하는 장면을 반복 블록을 사용해 만들기 • 순차구조 혹은 반복구조 알고리즘으로 ‘이상한 숲 속의 엔트리봇’ 미션8 해결 및 발표하기 	반복 나열, 패턴 이해, 알고리즘 (순차, 반복)

<표 III-9> EPL 교육 예시 지도안

주제	코딩으로 만드는 자기소개		회기 (시간)	EPL 3회기 (40분)
학습목표	코딩으로 자기소개 화면을 만들 수 있다.			
컴퓨팅 사고력	절차적 사고, 알고리즘(순차)			
단계 (시간)	교수-학습활동			자료
활동 과정 및 내용	도입 (7)	<ul style="list-style-type: none"> ● 전시학습 상기하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 5px; text-align: center;">엔트리로 자기 소개를 해보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		☞ 자기 소개 만들기 영상
	전개 (30)	<p><활동1> 자기 소개 하는 법 알기</p> <ul style="list-style-type: none"> - 스터디 학습하기로 들어가 목표 작품을 눌러 영상 확인하기 - ‘자기 소개 만들기’ 영상에 나온 오브젝트 찾아 열기 - 영상에 나온 장면을 기억하여 조립소에서 블록 찾아 조립하기 - 블록에 알맞은 대사 입력하기 <p><활동2> 나의 특징이 들어간 자기 소개 하기</p> <ul style="list-style-type: none"> - 내가 좋아하는 것, 나를 상징하는 이미지를 생각해 여러 가지 오브젝트 선택하고, 블록 조립하기 - 소개 글 3가지 이상 쓰기 - 엔트리로 자기 소개 발표하기 <p><활동3> 이상한 숲 속의 엔트리봇 미션 해결하기</p> <ul style="list-style-type: none"> - 이상한 숲 속의 엔트리봇 미션3 해결하기 - 자신이 세운 알고리즘 발표하기 		☞ 태블릿 PC, 자기 소개 만들기 영상 ☞ 태블릿 PC ☞ 태블릿 PC
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 학습내용 정리하고 알게 된 점, 어려웠던 점 이야기 나누기 		
	유의점	<ul style="list-style-type: none"> • 자신을 나타내는 오브젝트를 선택할 수 있게 하고 자신을 나타내는 알맞은 대사를 쓸 수 있도록 한다. • 어떤 블록을 조립해야 할지 스스로 탐구할 수 있도록 한다. 		

3) 효과적인 처치

효과적인 처치 구간은 중재가 끝난 10일 후 실시하였으며 두 가지 수업유형 중 EPL 교육이 수업 참여행동에서 상대적으로 효과가 높게 나타나 효과적인 처치에서는 EPL 교육을 실시하였다. 효과적인 처치 구간에서는 ‘나랑 놀자! 소프트웨어 입문편(발달장애학생용)’의 콘텐츠 5단원의 ‘엔트리, 터틀과 즐거운 상상놀이’ 중 3차시를 선정하였으며, 학생의 장애 정도와 학습 수준에 맞게 재구성하였다. 효과적인 처치 구간에서는 수업 참여행동을 측정하였다. 효과적인 처치 구간에서 실시한 교육 내용은 <표 III-10>과 같다.

<표 III-10> 효과적인 처치 구간의 교육 내용

차시	교육 내용
1차시	<조건을 생각하기> <ul style="list-style-type: none"> • ‘마우스를 클릭’ 하는 조건일 때 거미줄을 그리는 명령 표현하기 • 마녀가 하늘을 날아다니며 낙서를 하다가 ‘벽에 닿았을 때’ 랜덤으로 색이 바뀌는 명령 표현하기
2차시	<신비한 블록과 놀기> <ul style="list-style-type: none"> • ‘도장 찍기’ 블록으로 선인장을 클릭할 때마다 반복해서 심어지는 명령 표현하기 • 소리와 관련된 블록을 사용해 악기를 클릭했을 때 악기 소리가 나오는 명령 표현하기
3차시	<똑똑한 블록과 놀기> <ul style="list-style-type: none"> • 사용자의 대답을 입력받는 ‘대답’ 블록을 이용해 퀴즈를 내고 정답을 확인하는 명령 표현하기

나. 연구 기간

이 연구의 실험은 2022년 4월부터 7월까지 평일 방과 후에 실시하였다. 매주 화요일, 목요일 주 2회로 정하였으나, 연구참여자의 방과 후 개인 일정 참석, 병결로 인한 결석으로 수업결손이 생긴 경우 전체 참여자가 참석 가능한 요일로 대체하여 평균 주2~3회 실시하였다. 연구참여자 3명이 매회기 함께 수업을 진행하였으며 초등학교 1교시 수업 시간 기준으로 매회기 40분씩 수업하였고 1

일 1회기를 실시하였다. 기초선 3회, 중재 16회(언플러그드 교육 8회, EPL 교육 8회)를 실시하였으며, 중재 종료 10일 후 효과적인 처치 구간을 적용하여 3회기를 실시하였다. 기초선 회기가 끝난 후 컴퓨팅 사고력 사전 검사를 실시하였고 중재 회기가 끝난 후 컴퓨팅 사고력 사후 검사를 하였으며 검사 시간도 동일하게 40분으로 하였다.

다. 실험 장소

전체 실험 기간 동안 연구참여자가 재학하는 초등학교의 특수학급 교실에서 연구자가 방과 후에 실험을 진행하였다. 교실의 앞쪽에 스마트폰 3대를 설치하여 각 학생별로 동영상을 촬영하였다. 컴퓨팅 사고력을 측정하기 위한 사전, 사후 검사도 동일한 교실에서 진행하였다.

라. 중재 충실도

이 연구의 중재가 일관되게 이루어졌는지 알아보기 위해 중재 충실도 평가를 실시하였다. 중재 충실도 체크리스트는 윤혁(2016)과 김시원(2014)의 중재 충실도 양식을 이 연구에 맞게 수정하였다([부록5] 참조). 평가항목은 중재준비, 중재절차, 중재성실도로 구성하였다. 중재준비 영역에서는 행동 관찰을 위한 동영상 촬영 준비가 되었는지, EPL 교육에서 사용하는 태블릿 PC와 언플러그드 교육에서 사용하는 학습준비물이 준비되었는지 평가하도록 하였다. 중재절차 영역에서는 이 연구의 회기별 교육활동과 지도안을 제시하여 회기 순서에 맞게 중재 프로그램이 수행되었는지 평가하도록 하였고, 중재 중 학생에게 시범을 보였는지, 지도안에 맞게 활동이 이루어졌는지, 활동 후 평가 시간이 주어졌는지 평가하도록 하였다. 중재성실도 영역에서는 학생의 결과물에 대한 적절한 반응을 제공했는지, 반성적 사고를 위해 최소한의 촉진을 제공했는지, 각 활동별 충분한 시간을 제공했는지, 각 학생별로 80% 이상 활동에 참여하도록 하였는지 평가하도록 하였다. 각 문항별로 실행여부를 예/아니오로 평가되도록 제작하였다. <표 III-11>에 중재 충실도의 측정 항목을 간단히 요약하였다.

<표 III-11> 중재 충실도 측정 항목

영역	내용	문항수
중재준비	- 행동관찰을 위한 동영상 촬영 준비 - 태블릿 PC 또는 학습준비물 준비	2
중재절차	- 회기 순서에 맞는 중재 프로그램 수행 여부 - 시범 보이기 실행 여부 - 지도안 흐름에 따른 활동 수행 여부 - 활동 후 평가 시간 제공 여부	4
중재성실도	- 학생 결과물에 대한 적절한 반응 제공 여부 - 반성적 사고를 위한 최소한의 촉진 제공 여부 - 각 활동별 충분한 시간 제공 여부 - 각 학생별 80%이상 활동 참여 여부	4

중재 충실도를 측정하기 위해 연구참여자들의 동의를 받아 특수교육 경력 6년 차의 특수교사 1인과 교육경력 5년 차의 통합학급 담임교사 1인에게 녹화된 영상과 회기별 교육활동 및 지도안을 제공하고 중재가 충실히 진행되었는지 평가하도록 하였다. 평정 실시 전 중재 충실도 검사의 의의와 측정방법에 대해 간단한 연수를 실시하였다. 중재 16회기의 30%인 4회기를 측정하였으며 언플러그드 교육과 EPL 교육 중 무작위로 각각 2회씩 추출하여 평가하였다. 점수 산출은 실행해야 할 중재 요소를 실제로 실행한 중재 요소로 나누고 백분율로 구하여 산출하였다.

중재 충실도 결과는 언플러그드 교육 중재 충실도 95%(범위 90-100), EPL 교육 중재 충실도 97.5%(범위 90-100)로이며, 두 중재를 종합하면 중재 충실도는 총 평균 96.25%(범위 90-100)로 나타났다. 자세한 평균 점수와 범위는 <표 III-12>와 같다.

<표 III-12> 수업별 중재 충실도의 평균과 범위

영역	평균	범위	측정횟수(회)
언플러그드 교육	95	90-100	4
EPL 교육	97.5	90-100	4
계	96.25	90-100	8

마. 사회적 타당도

이 연구의 사회적 타당도를 측정하기 위해 김시원(2014)과 민종선, 김우리(2021) 연구의 사회적 타당도 문항을 이 연구에 맞게 수정하였다([부록6] 참조). 평가항목은 중재목표의 중요성, 중재의 수용성, 중재결과의 중요성으로 구성하였고, 각 문항별로 5점 척도로 평가되도록 제작하였다. 중재 목표의 중요성에서는 장애학생에게 SW교육이 필요하다고 생각되는지 평가하도록 하였다. 중재의 수용성에서는 장애학생이 SW교육을 어렵지 않게 수행할 수 있을지, 다른 교사에게도 추천하고 싶은지, SW교육이 장애학생에게 적절한지 평가하도록 하였다. 중재 결과의 중요성에서는 SW교육이 장애학생에게 유익한지 평가하도록 하였다.

사회적 타당도의 측정은 각 연구참여자의 통합학급 담임교사 3인이 평가하였다. 평정 시행 전 이 연구의 목적과 교육 프로그램의 내용에 대해 간단한 연수를 시행하였다. 사회적 타당도 결과는 5점 만점에서 총 평균 4.92점(범위 4.01-5.0) 퍼센트는 98.66%로 나타났다. 하위 항목별로 살펴보면 중재목표의 중요성은 평균 5점(범위 5.0-5.0)으로 100%로 나타났다. 중재의 수용성은 3가지 항목 중 첫 번째 항목은 평균 4.6점(범위 4.0-5.0)으로 92%, 두 번째와 세 번째 항목은 평균 5점(범위 5.0-5.0)으로 100%로 나타나 중재의 수용성은 총 97.33%로 나타났다. 중재결과의 중요성은 평균 5점(범위 5.0-5.0)으로 100%로 나타나 통합학급 교사들은 장애학생을 대상으로 실시하는 SW교육에 대하여 중재목표의 중요성과 중재결과의 중요성을 높게 인식하는 것으로 나타났다. 자세한 평균 점수와 범위는 <표 III-13>과 같다.

<표 III-13> 사회적 타당도 평균과 범위

영역	평균	범위	퍼센트(%)
중재목표의 중요성	5.0	5.0-5.0	100
중재의 수용성	4.6	4.0-5.0	92
	5.0	5.0-5.0	100
	5.0	5.0-5.0	100
중재결과의 중요성	5.0	5.0-5.0	100
계	4.92	4.0-5.0	98.66

4. 종속 변인

가. 문제해결력

1) 문제해결력 측정 영역

이 연구에서 문제해결력이란 SW교육에서 제시된 문제들을 알고리즘을 통해 해결하는 능력을 말한다. 장애학생의 SW교육 문제해결력을 측정하기 위해 지도교수 1인과 B교육대학교 컴퓨터교육과를 졸업하였고 5~6학년 「실과」 교과의 SW교육을 가르친 경험이 있는 교육경력 6년차인 초등교사 1인에게 자문을 구하여 진행하였다. 먼저 SW교육 문제해결력과 관련된 선행연구를 분석하여 문제해결력 측정 방법에 대해 논의하였다. 선행연구를 살펴보면 SW교육에서 이루어지는 문제해결력은 학업성취도 국제비교조사(PISA)의 문제해결 영역의 공개 문항 검사지(이은경, 이영준, 2007; 조성환 외, 2008), 자기보고식 문제해결력 검사지(김돈정, 2016; 김인호, 유미현, 2019; 서영호, 김종훈, 2018), 학생 관찰 평가(남수정, 2020; Dietz et al., 2019)가 주로 사용되었다. 검사지를 사용한 측정 도구들은 장애학생의 SW교육 문제해결을 측정하기에는 난도가 높거나 장애학생의 SW교육 문제해결력에서 나타나는 정확한 구인을 측정한다고 보기 어렵다. 따라서 논의를 통해 학생 관찰을 통한 방법을 선정하였다.

관찰 측정 영역에 대한 선행연구를 살펴보면 Dietz et al.(2019)의 연구에서는 쌓기나무를 통해 프로그래밍 문제해결 능력을 측정하였는데 10분간 과제 성공률, 완료 시간, 도전 횟수를 측정하였다. 정상목(2020)의 연구에서 소개한 EPL 문제해결력 평가 프로그램 온라인 저지(online judge)에서는 학생이 설계한 알고리즘의 실행 시간, 알고리즘의 정확성, 알고리즘의 효율성을 측정하였다. 또한 남수정(2020)의 연구에서는 문제해결 활동의 분석 기준을 알고리즘적 사고와 반성적 사고로 분류하고 학생들이 엔트리 활동을 통해 나타나는 문제해결 활동의 특성을 분석하였다. 알고리즘적 사고에서는 순차, 반복, 선택의 수행이 이루어졌는지, 반성적 사고에서는 검토(오류 수정), 발전(세련된 방법 찾기), 창의의 수행이 이루어졌는지 측정하였다. 이와 같은 선행연구를 바탕으로 이 연구의 SW교육 문제해결력 측정 영역은 도전 횟수, 정답 도달 시간, 오류점검

횟수, 효율적인 알고리즘 구상 영역이 1차로 선정되었으며, 2차 회의에서는 정답 도달 시간 영역과 관련하여 선행연구들이 비장애학생들을 대상으로 이루어진 연구이므로 정답에 도달할 것이라는 기본 전제를 두고 연구가 이루어진 측면이 있어 인지적 발달이 상대적으로 느린 지적장애 초등학생을 대상으로 한 연구에서는 정답에 도달하지 못하는 경우도 예상해야 한다는 지적이 나왔다. 또한 SW교육에서 문제해결은 답을 원하는 것보다 답을 해결하는 방법을 찾길 원하므로(한병래, 2013) 정답 도달과 상관없이 답을 해결하는 방법을 구상하는 부분을 측정할 수 있는 영역이 필요하다는 지도 교수와 자문 교사의 의견을 수렴하여 정상목(2020)과 Dietz et al.(2019)의 연구에서 제시한 실행 시간 영역과 10분간 과제 성공률 영역을 수정하여 1회 평균 구상 시간 영역을 추가하였다.

이 연구에서 측정한 문제해결 영역 중 도전 횟수란 알고리즘 문제를 제시하고 정답에 도달하기까지 도전한 횟수를 말한다. SW교육 문제해결은 일반적인 문제해결과 달리 답을 원하는 것보다 답을 해결하는 방법을 찾길 원한다(한병래, 2013). 따라서 알고리즘의 정확성은 답을 찾아가는 수행 횟수와 연결(김세용, 2002)되는 것이므로 이 영역에서는 한 문제당 도전 횟수를 빈도기록법을 이용해 측정하였다.

정답 도달 시간이란 한 문제 당 정답에 도달한 시간을 말한다. SW교육 문제해결에서는 한 번에 정답을 맞힐 수도 있지만 여러 번의 도전을 통해 답을 찾아가는 것이다. 이 영역에서는 문제 제시 후 정답에 도달할 때까지의 시간을 측정하였으며 Dietz et al.(2019)의 연구와 마찬가지로 한 문제당 10분의 제한을 두었다. 정답을 맞춘 시각에서 문제를 풀기 시작한 시각을 뺀 시간을 구하였다.

1회 평균 구상 시간이란 문제를 확인한 후 목적지까지의 알고리즘을 구상하고, 자신이 구상한 알고리즘을 명령어 카드 혹은 명령어 블록으로 나열하기까지의 시간을 말한다. 특히 SW교육 문제해결은 바로 정답이 나타나기도 하지만 1회 이상의 도전이 이루어지므로 도전할 때 마다의 시간을 측정하고자 하였다. 따라서 1회 평균 구상 시간에서는 과제 확인 후 명령어 블록을 조립 완료하는 시점까지의 시간을 측정하여 매회 완료 시각에서 매회 시작 시각을 뺀 시간을 구한 후, 한 문제 당 1회 평균 구상 시간을 산출하였다.

오류점검 횟수란 자신이 세운 알고리즘이 오답이었을 때 자신이 설계한 알고리즘을 처음부터 끝까지 복기하여 틀린 부분만 찾는 행위의 횟수를 말하며 이

는 프로그래밍에서 디버깅 과정과 동일하다. 이 영역에서는 오답임을 확인한 후 오류점검 행위를 관찰하여 빈도기록법을 이용해 측정하였다.

효율적인 알고리즘 구상이란 효율성 및 단순화의 원리에 따라 반복문을 사용해 가능한 간단한 알고리즘을 설계하는 것을 말한다. 이 연구에서는 언플러그드 및 EPL 교육의 각 5~8번 문제에 반복구조 문제를 제시하였으며 순차구조 혹은 반복구조 중 하나를 택하여 문제를 풀도록 하였다. 이 영역에서는 반복구조 도전 횟수를 빈도기록법으로 측정한 후 반복구조문 도전 횟수를 전체 도전 횟수로 나눈 후 백분율을 구하여 측정하였다.

문제해결력 측정지는 조작적 정의에 맞춰 연구자가 제작하였으며 [부록3]에 제시하였다. 이 연구에서의 문제해결력 조작적 정의는 <표 III-14>와 같다.

<표 III-14> 문제해결력 조작적 정의

순	문제해결력	조작적 정의		참고
1	도전 횟수	정답이 나올 때까지 도전한 횟수		Dietz et al. (2019)
2	정답 도달 시간	첫 과제 제시부터 정답이 나올 때까지 걸린 시간		정상목(2020), Dietz et al. (2019)
3	1회 평균 구상 시간	언플러그드 교육	과제 확인부터 알고리즘판에 명령어 카드를 다 놓을 때까지의 시간	정상목(2020), Dietz et al. (2019) 연구 수정·보완
		EPL 교육	과제 확인부터 블록조립소에서 코딩 블록을 조립한 후 시작 버튼을 눌러 기 전까지의 시간	
4	오류점검 횟수	교사의 도움 없이 문제에 대한 답을 처음부터 끝까지 검토하며 오류 부분만 찾아 고치는 횟수		남수정(2020)
5	효율적 알고리즘 구상	언플러그드 교육	반복의 의미를 이해하고 반복 명령어 카드를 한 개 이상 사용해 정답이 나올 때까지 사용한 비율	남수정(2020), 정상목(2020)
		EPL 교육	반복의 의미를 이해하고 반복 블록 한 개 이상 조립하여 정답이 나올 때까지 사용한 비율	

2) 문제해결력 측정 도구

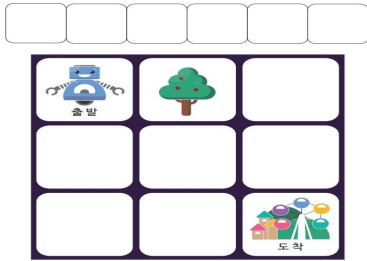
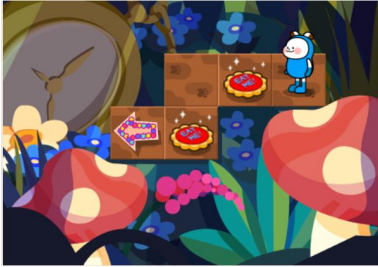
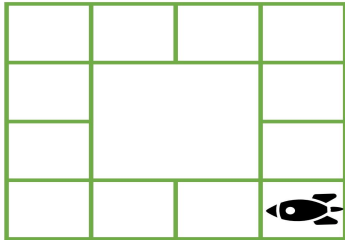
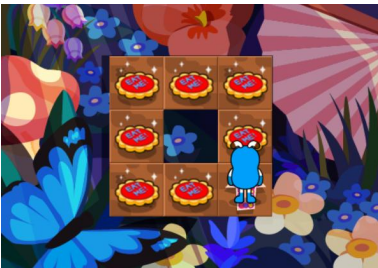
이 연구에서는 SW교육에서 나타난 문제해결력을 측정하기 위해 매회기마다 알고리즘 문제해결 활동을 제시하였으며, 알고리즘 문제를 해결하는 과정을 관찰하여 문제해결력을 측정하였다. 알고리즘 문제들은 지도교수 1인 및 B교육대학교 컴퓨터교육과를 졸업하였고 5~6학년 「실과」 교과의 SW교육을 가르친 경험이 있는 교육경력 6년차인 초등교사 1인에게 자문을 받아 언플러그드 교육 8문제, EPL 교육 8문제, 총 16문제를 선정 및 제작하였다. 1차 회의를 통해 알고리즘 문제는 모두 출발점에서 목적지까지 이동하는 문제로 구성하였으며, 연구참여자의 수준을 고려하여 프로그래밍 난이도가 낮은 순차문과 반복문으로 해결할 수 있는 문제들로만 구성하였다. 이철현(2019)의 연구에 따르면 순차, 반복, 선택구조문 중 선택구조는 특정한 조건을 구성해야 하므로 논리적 복잡성이 증가하여 학습자가 어려움을 크게 느끼는 것으로 나타났다. 따라서 추상적이고 복잡성이 덜한 순차와 반복 구조문만으로 알고리즘 문제해결력을 측정하고자 하였다.

언플러그드 교육의 알고리즘 문제는 국립특수교육원 웹사이트에서 배포한 ‘나랑 놀자! 소프트웨어 입문편(발달장애학생용)’의 1단원의 ‘롤러코스터를 찾아라’ 차시에서 제시한 4문제와 연구자 제작 4문제 총 8문제를 제시하였다. ‘롤러코스터를 찾아라’ 차시는 로봇이 롤러코스터를 타기 위해 길을 찾아가는 알고리즘 문제를 제공하며 15-1, 15-2, 15-3, 15-4의 4개 차시로 구성되어 있다. 15-1과 15-2차시에서는 순차문 구조의 문제, 15-3과 15-4 차시에서는 반복문 구조의 문제를 제공한다. 연구자 제작 문제는 EPL 교육의 알고리즘 문제와 난이도를 맞추기 위해 자문 교사와 2차 회의를 거쳐 문제를 제작하였으며 순차문 구조 2문제, 반복문 구조 2문제를 제작하였다.

다음으로 EPL 교육의 알고리즘 문제는 엔트리 사이트(<http://playentry.org>)에서 제공하는 ‘이상한 숲 속의 엔트리봇’의 미션1~8 문제를 선정하였다. ‘이상한 숲 속의 엔트리봇’이란 엔트리를 처음 접하는 이용자들에게 순차문, 반복문, 조건문, 함수문을 학습할 수 있도록 미션 형식으로 알고리즘 문제를 해결할 수 있게 하며 문제가 계속될수록 난도는 높아진다. 이 연구에서는 순차문, 반복문에 해당하는 문제만을 선정하였다.









언플러그드 교육과 EPL 교육에서 제시한 알고리즘 문제는 1~4번은 순차구조 문, 5~8번은 순차구조문 혹은 반복구조문으로 해결할 수 있도록 구성되어 있다. 언플러그드 교육과 EPL 교육의 알고리즘 문제들은 1~4번 문제는 순차구조로만 해결할 수 있게 하였고, 5~8번 문제는 단순화의 원리에 따라 간단한 문제해결 방법을 구상했는지 알아보기 위해 반복구조를 도입하였다. [그림 III-1]은 언플러그드 교육과 EPL 교육의 알고리즘 문제 예시이며, 각 문제에서 제시한 명령어 카드 및 명령어 블록은 [그림 III-2]와 같다.

[그림 III-1] 알고리즘 문제 예시

문제번호	언플러그드 교육	EPL 교육
문제2		
	나랑놀자! 소프트웨어 입문편 교재 문제	엔트리 문제
문제6		
	연구자 제작 문제	엔트리 문제

출처: <https://playentry.org/> (Copyright © NAVER Connect Foundation. Some Rights Reserved.)

[그림 III-2] 명령어 카드 및 명령어 블록

수업유형		명령어 카드 및 명령어 블록		
언플러그드 교육	‘나랑 놀자 소프트웨어’ 교재 명령어 카드	 오른쪽 한 칸 이동	 아래로 한 칸 이동	 반복하기
	연구자 제작 명령어 카드			 
EPL 교육	엔트리 명령어 블록			

나. 수업 참여행동

수업 참여행동이란 수업에서 이루어지는 과정에서 관찰되는 행동(변관석, 2016)으로 연구마다 연구참여자의 특성에 맞게 정의와 요소를 달리하였다. 이 연구에서는 선행연구 및 학습자의 특성에 맞추어 수업 참여행동을 자발적 질문, 주의집중 행동, 과제회피 행동으로 정의하였으며, 수업전반에서 나타나는 행동들을 관찰하여 측정하였다.

자발적 질문이란 수업 상황 중 수업 내용에 대해 모르는 부분을 교사에게 물어보는 행동을 말한다. 자발적 질문은 수업 전체 시간을 관찰하여 학생들이 질문한 행동을 빈도기록법으로 측정하였다([부록4] 참고). 주의집중 행동이란 교사 혹은 발표하고 있는 학생을 5초 이상 마주치는 행동과 과제를 수행할 때 과

제에 5초 이상 집중하는 행동으로 정의하였다. 과제회피 행동이란 선행연구(고정현, 이한얼, 2015; 임해주, 2017)와 연구참여자의 특성에 맞추어 설정하였으며 자리 이탈하기, 수업과 관련 없는 물건 만지거나 신체적 행동 반복하기, 과제 수행요구에도 가만히 앉아 있기 등의 행동이 5초 이상 발생하는 것으로 정의하였다. 주의집중과 과제회피 행동의 측정지는 이상철(2012), 조은빛(2022) 및 조미현(2012)의 연구를 참고하여 수정·보완하여 측정하였다([부록4] 참고). 수업 중 전개 부분에 해당하는 30분을 측정하였으며, 1분당 3개(총 90개)의 간격으로 나누고 15초 관찰, 5초간 기록하는 간격기록법으로 측정하였다. 시간 간격 내에 주의집중 행동과 과제회피 행동이 5초 이상 발생하면 해당 구간에 발생 표시(+)를 하였고, 점수 산출은 총 간격을 행동발생 간격 수로 나누고 백분율로 구하여 산출하였다. 수업 참여행동의 조작적 정의는 <표 III-15>와 같다.

<표 III-15> 수업 참여행동 조작적 정의

순	수업 참여행동	조작적 정의	참고
1	자발적 질문	수업 내용에 대해 모르는 부분을 물어보는 행동	고정현, 이한얼(2015), 김태용, 최하영(2013)
2	주의집중 행동	발화자(교사 혹은 학생)의 시선을 5초 이상 마주치고, 과제를 수행할 때 5초 이상 집중하는 행동	김태용, 최하영(2013) 이선영, 강옥려(2021)
3	과제회피 행동	5초 이상 자리 이탈하기, 수업과 관련 없는 물건 만지기, 수업과 관련 없는 신체적인 행동 반복하기, 과제 수행 요구 시 가만히 앉아 있기	고정현, 이한얼(2015), 임해주(2017)

다. 컴퓨팅 사고력

1) 컴퓨팅 사고력 측정 도구

컴퓨팅 사고력은 SW교육을 통해 습득한 지식을 바탕으로 문제를 해결하는 역량을 의미한다. 컴퓨팅 사고력은 단일한 하나의 역량으로 측정되는 것이 아니

며 컴퓨팅 사고력을 구성하는 하위 역량들을 측정하여 그 효과성을 측정할 수 있다. 컴퓨팅 사고력의 측정방법은 크게 3가지로 나눌 수 있다(김민정 외, 2017). 첫째, 특정 검사도구를 사용하는 경우(김거현, 유인환, 2017; 노지예, 이정민, 2018; 문우중 외, 2020; 박선주, 2019; 윤선희, 2018; 이재호, 김선향, 2021; 이정민 외, 2018a), 둘째, 설문조사를 실시하는 경우(노지예, 이정민, 2017; 노지예, 이정민, 2018; 오지훈 외, 2021; 이정민 외, 2018b), 셋째, 관찰방법(유인환, 2016)을 사용하는 경우이다. 이 연구에서는 첫 번째 유형에 속하는 특정 검사도구를 사용하여 컴퓨팅 사고력을 측정하고자 하였으며 그 중 비버챌린지(babras challenge) 기출문제를 선정하였다.

비버챌린지의 과제들은 정보과학의 개념으로 구성되어있으며, 추상화, 알고리즘적 사고, 프로그래밍 능력 등 컴퓨팅 사고력을 측정할 수 있는 모듈형 과제들로 구성되어 있다. 비버챌린지 문제는 70여개 국가가 공동 개발하며 특정 소프트웨어나 하드웨어 및 프로그래밍에 대한 특별한 사전 지식이 없어도 문제를 풀 수 있고 학생의 연령과 수준을 구분하여 과제를 제시하고 있다(박선주, 2019; 정웅열, 이영준, 2017; Román-González et al., 2017). 비버챌린지는 컴퓨팅 사고력을 진단하며 보편교육을 지향하는 평가 모델로 알려져 있다(정웅열, 이영준, 2017). 선행연구를 살펴보면 초등학생 혹은 초등 저학년들을 대상으로 컴퓨팅 사고력의 효과를 측정하기 위해 여러 연구에서 비버챌린지를 사용하였으며, 이를 통해 컴퓨팅 사고력의 효과를 입증하였다(강은희, 2019; 강은희, 2018; 김춘호, 2022; 노지예, 이정민, 2018; 박선주, 2019; 송옥지 외, 2020; 조은진, 2022). 특히 각 학년별로 난이도를 고려하여 문제가 출제되기 때문에 초등학교 저학년의 문제는 지적장애 초등학생들도 풀 수 있을 것으로 예상된다. 사전 검사와 사후 검사를 동일한 검사지를 활용하였는데 학습효과를 방지하기 위해 사전 검사 후 정답을 알려주지 않았으며 약 3개월 후 사후 검사를 실시하였기 때문에 사전 학습효과는 거의 나타나지 않을 것으로 판단하였다.

2) 컴퓨팅 사고력 측정 영역

각 문항의 컴퓨팅 사고력 세부 구성 요소는 호주 비버챌린지 사이트(<http://bebras.edu.au>)에서 제공하는 해결안내서(solutions guide)에서 제시한 요소를

적용하였다. 비버첼린지 해결안내서에 제시한 컴퓨팅 사고력의 하위 요소는 문제 분해, 패턴인식, 추상화, 모델링 및 시뮬레이션, 알고리즘, 평가 6가지이다. 첫째, 문제 분해란 주어진 문제를 해결 가능한 수준의 작은 문제로 쪼개는 것을 의미한다. 문제 해결 계획을 세울 때 필요한 요소로서 자신이 해결할 수 있는 부분과 해결하지 못하는 부분, 필요한 요소와 불필요한 요소를 나누어 보는 것도 문제 분해 과정에 포함된다. 둘째, 패턴인식이란 문제를 분해하여 반복되거나 비슷한 특성을 가진 일정한 규칙을 탐색하는 능력이다. 셋째, 추상화란 패턴인식을 통해 발견한 공식이나 원리를 설정하여 문제해결에서 반드시 필요한 핵심요소를 찾고 반복되는 패턴의 복잡함을 단순화시키는 능력이며 핵심요소를 추출하는 것은 불필요한 요소를 없애고 문제에 필수적인 요소만 남기는 것을 의미한다. 넷째, 모델링 및 시뮬레이션이란 문제를 해결하기 위한 절차를 모형화하여 가상으로 실험해보는 과정이다. 다섯째, 알고리즘이란 문제를 해결하기 위한 순서와 방법을 나열한 것을 말한다. 문제를 해결하기 위해 순서와 방법을 나열한 것으로서 추상화된 핵심 원리를 절차화된 순서에 삽입하여 완성된다. 알고리즘을 표현하는 대표적인 방법은 순서도와 프로그래밍 언어 등이 있다. 여섯째, 평가란 문제를 해결한 결과를 평가하는 것으로서 문제를 정확히 해결했는지 검증하는 능력이다(교육부, 2018). 이 연구에서 측정하는 컴퓨팅 사고력 세부 구성 요소를 요약한 내용은 <표 III-16>과 같다.

<표 III-16> 컴퓨팅 사고력 세부 요소

컴퓨팅 사고력 기술	내용
문제 분해 (decomposition)	문제를 더 잘게, 더 쉽게 부분으로 쪼개기
패턴인식 (pattern recognition)	데이터를 분석하고 데이터를 이해하기 위한 패턴 찾기
추상화 (abstraction)	유용한 정보를 찾고 도움이 되지 않는 정보 제거하기
모델링 및 시뮬레이션 (modelling and simulation)	문제를 해결하기 위해 여러 가지 해결책을 시도하거나 정보 경로를 추적하기
알고리즘 (algorithms)	문제를 해결하기 위해 순서화된 단계 만들기
평가 (evaluation)	문제에 대한 해결책을 평가하고 그 정보를 새로운 문제에 다시 사용하기

출처: <http://bebras.edu.au>

이 연구에서는 2019년에 발행된 비버챌린지 2019년도 기출문제집(한국비버챌린지, 2019)의 문제 중 연구대상자에게 적합한 초등학교 1~2학년에 해당하는 8문제를 제시하였다. 이 연구의 연구참여자의 국어 수준은 2022학년도 IEP에 준하여 1학년 2학기~2학년 1학기에 해당하며 짧은 문장을 읽고 정확하지는 않지만 70% 정도로 의미를 파악하는 수준이 되기 때문에 초등학교 1~2학년 문제가 적합하다고 보았다. 2019년 초등학교 1~2학년 기출문제는 5지 선다의 객관식 6문제와 주관식 2문제로 이루어져 있다. 비버챌린지는 문항별, 정답 유무별로 배점이 다르지만, 이 연구에서는 각 문항 당 정답은 1점, 오답은 0점으로 설정하였다. 따라서 이 연구의 컴퓨팅 사고력은 점수가 클수록 컴퓨팅 사고력이 높은 것이며 만점은 8점이다.

각 문제별 내용과 측정하는 컴퓨팅 사고력 요소는 다음과 같다. 1번 문제는 워터파크에 입장 가능한 나이를 알려주는 조건문을 보고 남매의 나이와 비교한 후 남매가 워터파크에 입장할 수 있을지 없을지를 유추하는 문제이다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 추상화, 알고리즘을 측정한다. 2번 문제는 로봇 청소기가 가장 마지막에 줍는 쓰레기가 무엇인지 찾는 문제로, 로봇 청소기는 흠어진 쓰레기들 중 가장 가까운 쓰레기부터 이동하여 쓰레기를 줍는다는 조건이 제시된다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 추상화, 알고리즘을 측정한다. 3번 문제는 7개의 다른 양말을 나열한 후, 요일에 따라 신는 양말의 조건을 제시하여 토요일에 신는 양말을 유추하는 문제이다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 패턴인식, 추상화, 알고리즘을 측정한다. 4번 문제는 스크래치 아트 종이를 굵었을 때 3가지 색깔만 보이는 그림을 찾는 문제로, 스크래치 아트 종이 밑에 숨겨진 4가지 색의 위치를 알려주고 각 문항의 그림과 비교하여 3가지 색깔만 보일 수 있는 그림이 무엇일지 유추하는 문제이다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 패턴인식, 추상화를 측정한다. 5번 문제는 사탕 4개와 비버 4마리를 나열하고, 조건문에 따라 특정 사탕을 가져가는 비버의 이름을 선택하는 문제이다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 추상화를 측정한다. 6번 문제는 집과 학교까지의 지도를 제시하고 가장 빨리 도착할 수 있는 최단 경로를 찾는 문제이다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 추상화, 모델링 및 시뮬레이션, 알고리즘을 측정한다. 7번 문제는 무작위로 찍힌 알파벳과 나침반에 찍힌 알파벳을 비교하여 격자무늬 지도에 마지막 도착 지점을 표시하는 문제이다. 이 문제는 컴퓨팅

사고력 요소 중 문제 분해, 추상화, 모델링 및 시뮬레이션, 알고리즘을 측정한다. 8번 문제는 금액에 따라 모양이 다른 코인 5가지를 제시한 후, 13코인을 가능한 적은 개수로 선택하여 코인을 그리는 문제이다. 이 문제는 컴퓨팅 사고력 요소 중 문제 분해, 추상화, 알고리즘을 측정한다. 비버챌린지 문제 제목과 각 문항별 컴퓨팅 사고력 세부 구성요소를 정리한 표는 <표 III-17>과 같다.

<표 III-17> 비버챌린지 문항별 컴퓨팅 사고력 세부 구성 요소

문제 번호	문제 제목	컴퓨팅 사고력 세부 구성 요소
1	워터파크 입장	문제분해, 추상화, 알고리즘
2	잔디밭 청소	문제분해, 추상화, 알고리즘
3	FIFO 양말	문제분해, 패턴인식, 추상화, 알고리즘
4	스크래치 아트 종이	문제분해, 패턴인식, 추상화
5	사탕 가게	문제분해, 추상화
6	집으로 오는 길	문제분해, 추상화, 모델링 및 시뮬레이션, 알고리즘
7	숲속의 토끼	문제분해, 추상화, 모델링 및 시뮬레이션, 알고리즘
8	비버 코인	문제분해, 추상화, 알고리즘

라. 관찰자 간 신뢰도

관찰기록의 신뢰도를 검증하기 위해 연구자와 특수교육 경력 6년 차의 특수교사 1인이 녹화된 동영상을 분석하였다. 연구자는 분석교사에게 문제해결력과 수업 참여행동에 대한 관찰 및 평가 기준의 훈련을 제공하였으며 훈련의 결과가 90% 이상일 때 종속변인의 측정을 시작하였다. 문제해결력 측정지, 주의집중과 과제회피 측정지, 자발적 질문 측정지([부록3], [부록4] 참고)를 비교 분석하였다. 기초선, 중재, 효과적인 처치 총 22회기 중 30%인 6회기를 무작위로 선택하여 신뢰도를 측정하였다. 기초선 1회, 중재 4회, 효과적인 처치 1회를 선정하여 측정하였다. 관찰자 간 신뢰도는 매회 한 명을 선택해 측정하였으며 3명 중 무작위로 선택하였다. 관찰자 간 신뢰도는 일치된 수를 일치된 수와 불

일치된 수를 더한 수로 나누어 백분율로 구하여 산출하였다. 관찰자 간 신뢰도는 총 평균 98.16(범위 95-100)으로 나타났으며 하위 영역별로 살펴보면 문제 해결력은 평균 98%(범위 97-100), 수업 참여행동 중 주의집중과 과제회피 행동은 평균 96.5%(범위 95-100), 자발적 질문은 평균 100%(범위 100-100)로 나타났으며 자세한 평균 점수와 범위는 <표 III-18>과 같다.

<표 III-18> 관찰자 간 신뢰도의 평균과 범위

영역		평균	범위	측정횟수
문제해결력		98	97-100	4회
수업 참여 행동	자발적 질문	100	100-100	6회
	주의집중, 과제회피	96.5	95-100	6회
계		98.16	95-100	16회

IV. 연구 결과

1. 언플러그드 교육과 EPL 교육이 문제해결력에 미치는 효과 비교

이 연구에서는 언플러그드 교육과 EPL 교육에서 알고리즘 문제를 제시하고 해결하는 과정을 관찰하여 중재별로 문제해결력을 비교하고자 하였다. 언플러그드 교육에서 8문제, EPL 교육에서 8문제, 총 16문제를 해결하도록 하였다. 1~4번 문제는 반복구조가 도입되지 않은 순차구조의 문제이며, 5~8번 문제는 반복구조가 도입된 문제로 구성하였다. 수업유형별로 각 8가지 문제를 해결해 나가는 모습을 통해 문제해결력 즉, 도전 횟수, 정답 도달 시간, 1회 평균 구상 시간, 오류점검 횟수, 효율적인 알고리즘 구상에서 두 가지 교육 프로그램이 어떠한 차이를 나타내는지 알아보하고자 하였다. 전체 문제해결력을 종합하여 두 중재의 효과를 비교하고자 하였지만 각 영역별로 효과가 있는 수업유형이 다르게 나타나 각 영역별로 결과를 비교하고자 한다.

<표 IV-1> 전체 학생의 중재별 평균 문제해결력 결과

문제해결력	학생	중재			
		언플러그드 교육		EPL 교육	
		평균	(범위)	평균	(범위)
도전 횟수	학생A	3회	(1~8)	7.25회	(1~17)
	학생B	3.25회	(1~7)	4.87회	(1~9)
	학생C	2.5회	(1~6)	5.25회	(1~20)
정답 도달 시간	학생A	3분 3초	(01:04-07:49)	2분 5초	(00:23-05:55)
	학생B	3분 40초	(00:53-08:13)	2분 18초	(00:27-05:10)
	학생C	2분 27초	(00:30-05:03)	1분 31초	(00:09-08:12)
1회 평균 구상 시간	학생A	1분 22초	(00:32-02:18)	19초	(00:10-00:37)
	학생B	1분 11초	(00:27-02:08)	25초	(00:13-00:32)
	학생C	1분 35초	(00:30-03:36)	14초	(00:04-00:25)
오류점검 횟수	학생A	0.25회	(0~1)	0회	(0~0)
	학생B	0.125회	(0~1)	0회	(0~0)
	학생C	0회	(0~0)	0회	(0~0)
효율적인 알고리즘 구상	학생A	100%	(100~100)	56.25%	(20~100)
	학생B	88%	(67~100)	27.22%	(20~33)
	학생C	67%	(17~100)	75%	(33~100)

전체 학생의 중재별 평균 문제해결력 수치는 <표 IV-1>과 같다. 문제해결력의 평균 점수를 살펴보면 도전 횟수와 효율적인 알고리즘 구상 영역에서 언플러그드 교육이 EPL 교육 보다 효과적인 것으로 나타났다. 1회 평균 구상 시간과 정답 도달 시간은 EPL 교육이 언플러그드 교육보다 효과적인 것으로 나타났으며, 학생C의 경우 효율적인 알고리즘 구상 영역에서 EPL 교육이 효과적인 것으로 나타났다. 오류점검 횟수는 발생률이 현저하게 낮아 두 중재 간 우위적인 효과를 주장하기 어렵다.

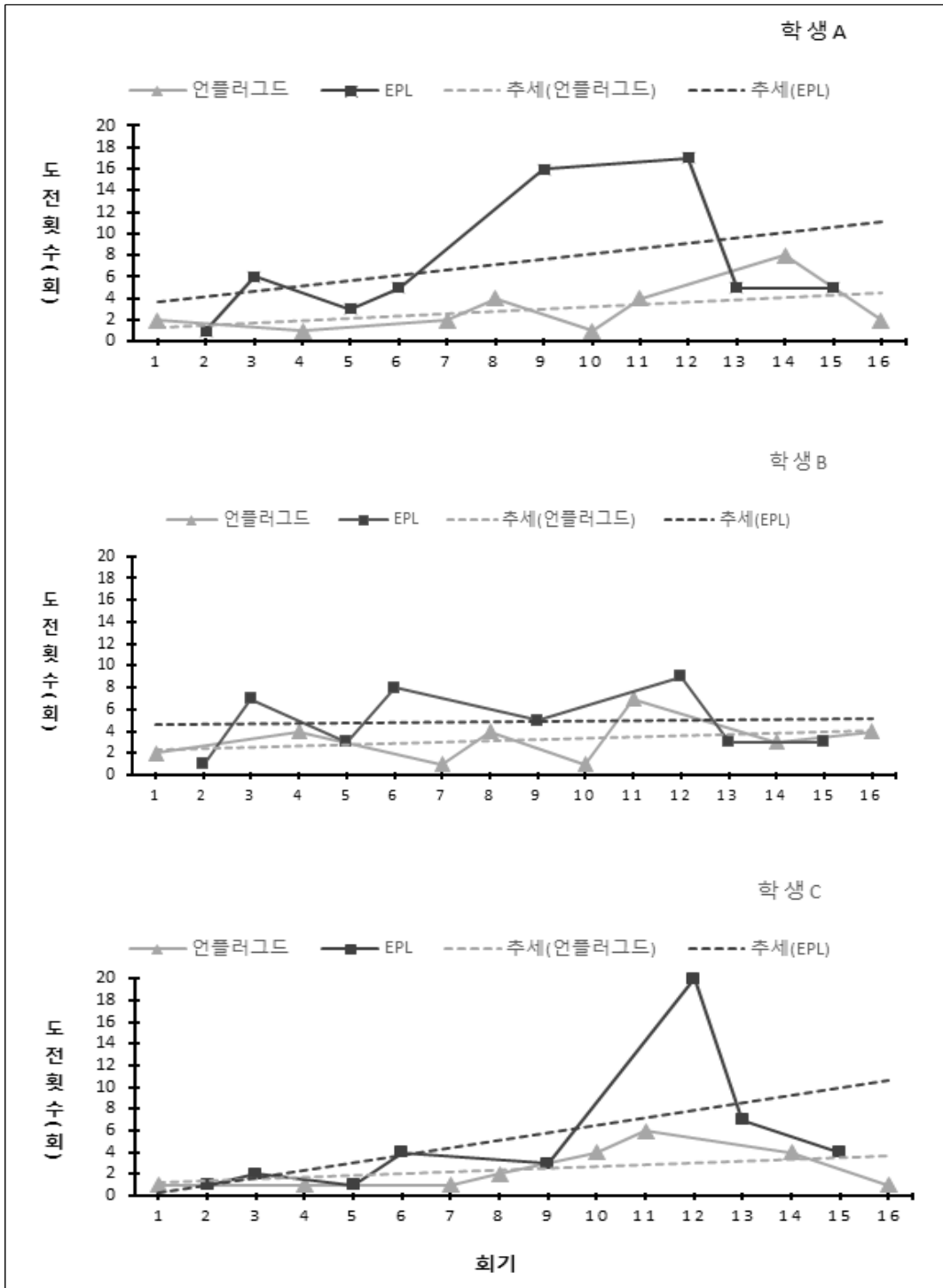
가. 도전 횟수 비교

도전 횟수는 알고리즘 문제를 제시했을 때 정답을 맞출 때까지 도전한 횟수이며, 적은 횟수로 정답에 도달하면 문제해결력이 높다고 볼 수 있다. 학생A와 학생C는 수업 시간의 제한으로 6번 문제에서 정답에 도달하지는 못했지만 제한 시간 전까지 도전한 횟수를 도전 횟수에 산정하였다. 알고리즘 문제의 해결을 통해 나타난 전체 학생의 문제별 도전 횟수와 평균은 <표 IV-2>, 그래프와 추세선은 [그림 IV-1]과 같다.

<표 IV-2> 수업유형별 도전 횟수 및 평균 (단위:회)

학생	수업 유형	문제번호								평균
		1	2	3	4	5	6	7	8	
A	언플러그드 교육	2	1	2	4	1	4	8	2	3
	EPL 교육	1	6	3	5	16	17	5	5	7.25
B	언플러그드 교육	2	4	1	4	1	7	3	4	3.25
	EPL 교육	1	7	3	8	5	9	3	3	4.87
C	언플러그드 교육	1	1	1	2	4	6	4	1	2.5
	EPL 교육	1	2	1	4	3	20	7	4	5.25

[그림 IV-1] 수업유형별 도전 횟수 그래프와 추세선



학생A의 평균 횡수를 살펴보면 언플러그드 교육 도전 횡수는 평균 3회(범위 1~8회)였으며, EPL 교육에서 도전 횡수는 평균 7.25회(범위 1~17회)로 언플러그드 교육이 EPL 교육보다 평균 4.25회 더 낮게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 초반에는 변화율이 거의 없다가 후반부에서 변화율이 크게 나타났으며 경향선은 우상향의 추세를 보였다. EPL 교육의 경우 반복구조 문제가 도입된 중반부에서 변화가 크게 나타났으며 후반부에서 횡수가 다시 줄어들었고 경향선은 우상향의 추세를 보였다.

학생B의 평균 횡수를 살펴보면 언플러그드 교육 도전 횡수는 평균 3.25회(범위 1~7회), EPL 교육에서 도전 횡수는 평균 4.87회(범위 1~9회)로 언플러그드 교육이 EPL 교육보다 평균 1.62회 더 낮게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 정답에 도달하지 못했던 6번 문제를 제외하고 횡수에 대한 큰 변화는 없었으며 안정적인 경향을 유지하였다. EPL 교육의 경우 초반 2번 문제부터 중반부까지 큰 변화를 보였으나 후반부로 갈수록 도전 횡수가 낮아지는 경향을 보였고 경향선은 약한 우상향의 추세를 보였다.

학생C의 평균 횡수를 살펴보면 언플러그드 교육 도전 횡수는 평균 2.5회(범위 1~6회), EPL 교육 평균 5.25회(범위 1~20회)로 언플러그드 교육이 EPL 교육보다 평균 2.75회 더 낮게 나타났다. 그래프를 살펴보면 반복문이 도입된 중반부부터 도전 횡수가 늘어나며 변화율이 크게 나타났고 경향선은 우상향의 추세를 보였다. EPL 교육의 경우 역시 반복문이 도입된 중반부부터 변화율이 크게 나타났고 경향선은 우상향 추세를 보였다.

이상에서 결과를 종합해볼 때 학생 3명 모두 언플러그드 교육의 알고리즘 문제를 해결할 때 도전 횡수가 낮게 나와 문제해결력의 도전 횡수 영역에서는 언플러그드 교육이 효과적인 것으로 나타났다.

나. 정답 도달 시간 비교

정답 도달 시간은 알고리즘 문제를 제시한 순간부터 정답을 맞출 때 까지 걸린 시간이며, 적은 시간으로 정답에 도달하면 문제해결력이 높다고 볼 수 있다. 학생A와 학생C는 수업 시간의 제한으로 6번 문제에서 정답에 도달하지는 못하였으므로 결과에 산정하지는 않았다. 알고리즘 문제해결을 통해 나타난 전

체 학생의 문제별 정답 도달 시간과 그 평균은 <표 IV-3>, 그래프와 추세선은 [그림 IV-2]와 같다.

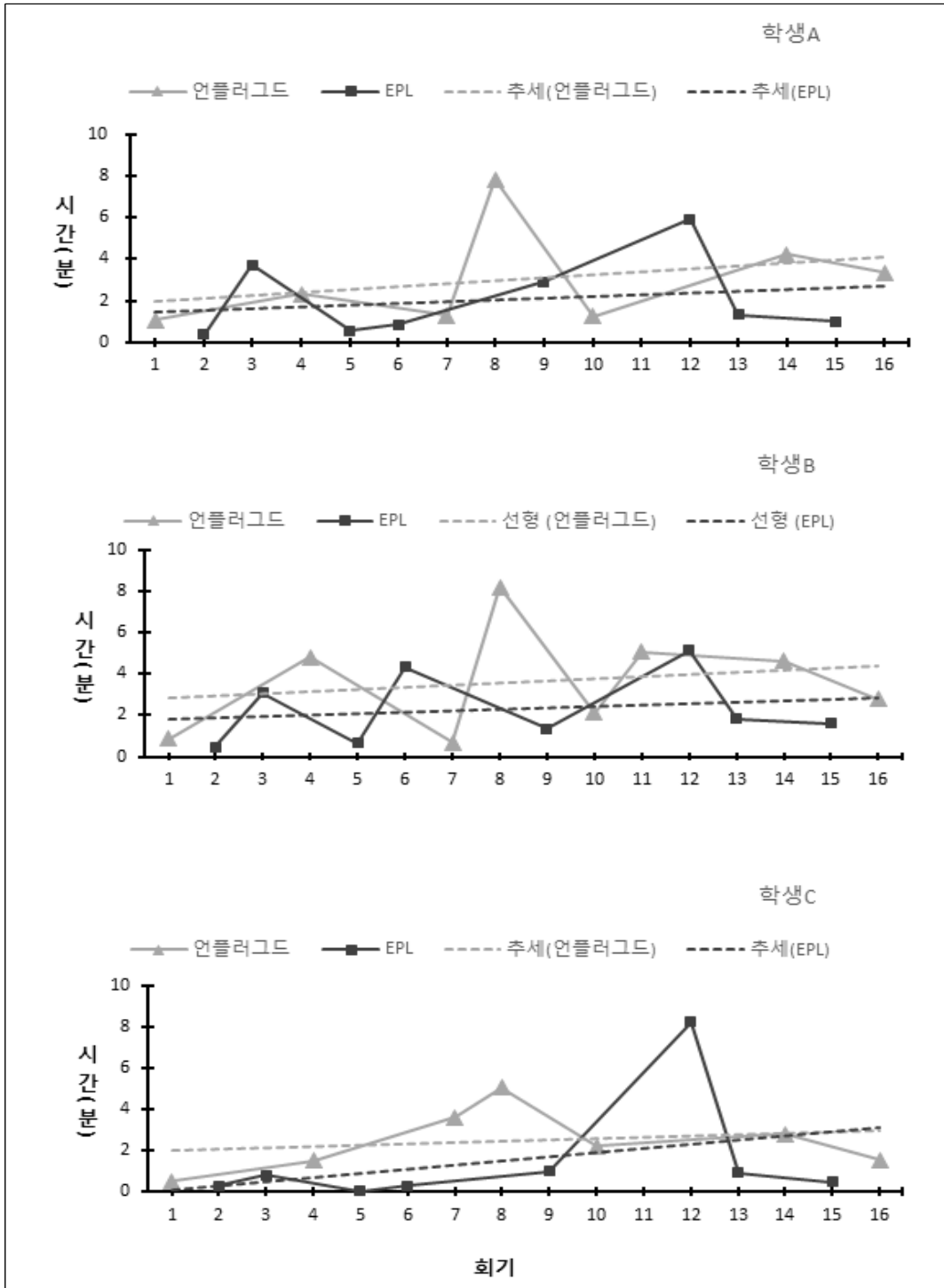
<표 IV-3> 수업유형별 정답 도달 시간 및 평균 (단위:분)

학생	수업 유형	문제번호								평균
		1	2	3	4	5	6	7	8	
A	언플러그드 교육	1:04	2:18	1:17	7:49	1:15	미도달	4:14	3:21	3:03
	EPL 교육	0:23	3:42	0:34	0:50	2:54	5:55	1:19	0:59	2:05
B	언플러그드 교육	0:53	4:50	0:42	8:13	2:08	5:05	4:39	2:47	3:40
	EPL 교육	0:27	3:04	0:39	4:20	1:22	5:10	1:49	1:37	2:18
C	언플러그드 교육	0:30	1:29	3:36	5:03	2:11	미도달	2:47	1:33	2:27
	EPL 교육	0:17	0:49	0:09	0:18	1:00	8:12	0:54	0:27	1:31

학생A의 정답 도달 시간을 살펴보면 언플러그드 교육 정답 도달 시간은 평균 3분 3초(범위 1:04~7:49)였으며, EPL 교육에서 정답 도달 시간은 평균 2분 5초(범위 0:23~5:55)로 EPL 교육이 언플러그드 교육보다 평균 58초 더 적게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 중반부부터 시간이 급격히 늘어났으며 후반부까지 오르내리는 변화율이 크게 나타났고 경향선은 우상향의 추세를 보였다. EPL 교육의 경우 초반부터 중반까지 오르내리는 변화율이 크게 나타났고 후반부부터 시간이 줄어드는 안정세를 보여 경향선은 안정적인 경향을 유지하였다.

학생B의 정답 도달 시간을 살펴보면 언플러그드 교육 정답 도달 시간은 평균 3분 40초(범위 0:53~8:13), EPL 교육 정답 도달 시간은 평균 2분 18초(범위 0:27~5:10)로 EPL 교육이 언플러그드 교육보다 평균 52초 더 낮게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 초반부터 중반부까지 오르내리는 변화율이 크게 나타났으며 후반부에는 높은 안정세를 보였고 경향선은 우상향의 추세를 보였다. EPL 교육의 경우 초반부터 중반부까지 오르내리는 변화율이 크게 나타났으며 후반부에는 낮은 안정세를 보여 경향선은 매우 약한 우상향의 추세를 나타냈다.

[그림 IV-2] 정답 도달 시간 그래프와 추세선



학생C의 정답 도달 시간을 살펴보면 언플러그드 교육에서 정답 도달 시간은 평균 2분 27초(범위 0:30~5:03), EPL 교육에서 정답 도달 시간은 1분 31초(범위 0:09~8:12)로 EPL 교육이 언플러그드 교육보다 평균 56초 더 낮게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 중반부부터 시간이 상승하였고 후반부에 낮은 안정세를 보였으며 경향선은 우상향의 추세를 보였다. EPL 교육의 경우 중반부까지 낮은 안정세를 보이다가 6번 문제에서 큰 오름세를 보였으며 그 이후로 낮은 안정세를 보였고 경향선은 우상향의 추세를 나타냈다.

이상에서 결과를 종합해볼 때 학생 3명 모두 EPL 교육의 알고리즘 문제를 해결할 때 정답 도달 시간이 적게 나타나 문제해결력의 정답 도달 시간 영역에서는 EPL 교육이 효과적인 것으로 나타났다.

다. 1회 평균 구상 시간 비교

1회 평균 구상 시간은 알고리즘 문제를 제시했을 때 문제를 보고 해결안을 구상하여 알고리즘을 세울 때까지의 시간을 말한다. 적은 시간으로 해결안을 구상하면 문제해결력이 높다고 볼 수 있다. 학생A와 학생C는 수업 시간의 제한으로 6번 문제에서 정답에 도달하지는 못했지만 제한 시간 전까지 도전한 횟수의 구상 시간을 측정하여 1회 평균 구상 시간에 산정하였다. 알고리즘 문제 해결을 통해 나타난 전체 학생의 문제별 1회 평균 구상 시간과 평균은 <표 IV-4>, 그래프와 추세선은 [그림 IV-3]과 같다.

학생A의 1회 평균 구상 시간을 살펴보면 언플러그드 교육 1회 평균 구상 시간은 평균 1분 22초(범위 0:32~2:18)였으며, EPL 교육에서 1회 평균 구상 시간은 평균 19초(범위 0:10~0:37)로 EPL 교육이 언플러그드 교육보다 평균 1분 3초 더 적게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 짝수번호의 문제마다 1회 평균 구상 시간이 늘어났지만 경향선은 안정적인 추세를 유지하였다. EPL 교육의 경우 변화율이 거의 없는 안정되는 양상을 보여주었고 경향선 역시 안정적인 추세를 유지하였다.

학생B의 1회 평균 구상 시간을 살펴보면 언플러그드 교육 1회 평균 구상 시간은 평균 1분 11초(범위 0:27~2:08), EPL 교육 1회 평균 구상 시간은 평균 19초(범위 0:13~0:32)로 EPL 교육이 언플러그드 교육보다 평균 52초 더 낮게 나

타났다. 그래프를 살펴보면 초반에는 적은 시간으로 안정세를 보이다가 중반부에 큰 변화율이 나타났으며 후반부에 다시 초반과 같아졌고 경향선은 미세한 우상향의 추세를 보였다. EPL 교육의 경우 전반적으로 안정적인 경향을 유지하였으며 경향선 역시 낮은 수준에서 안정적인 추세를 보였다.

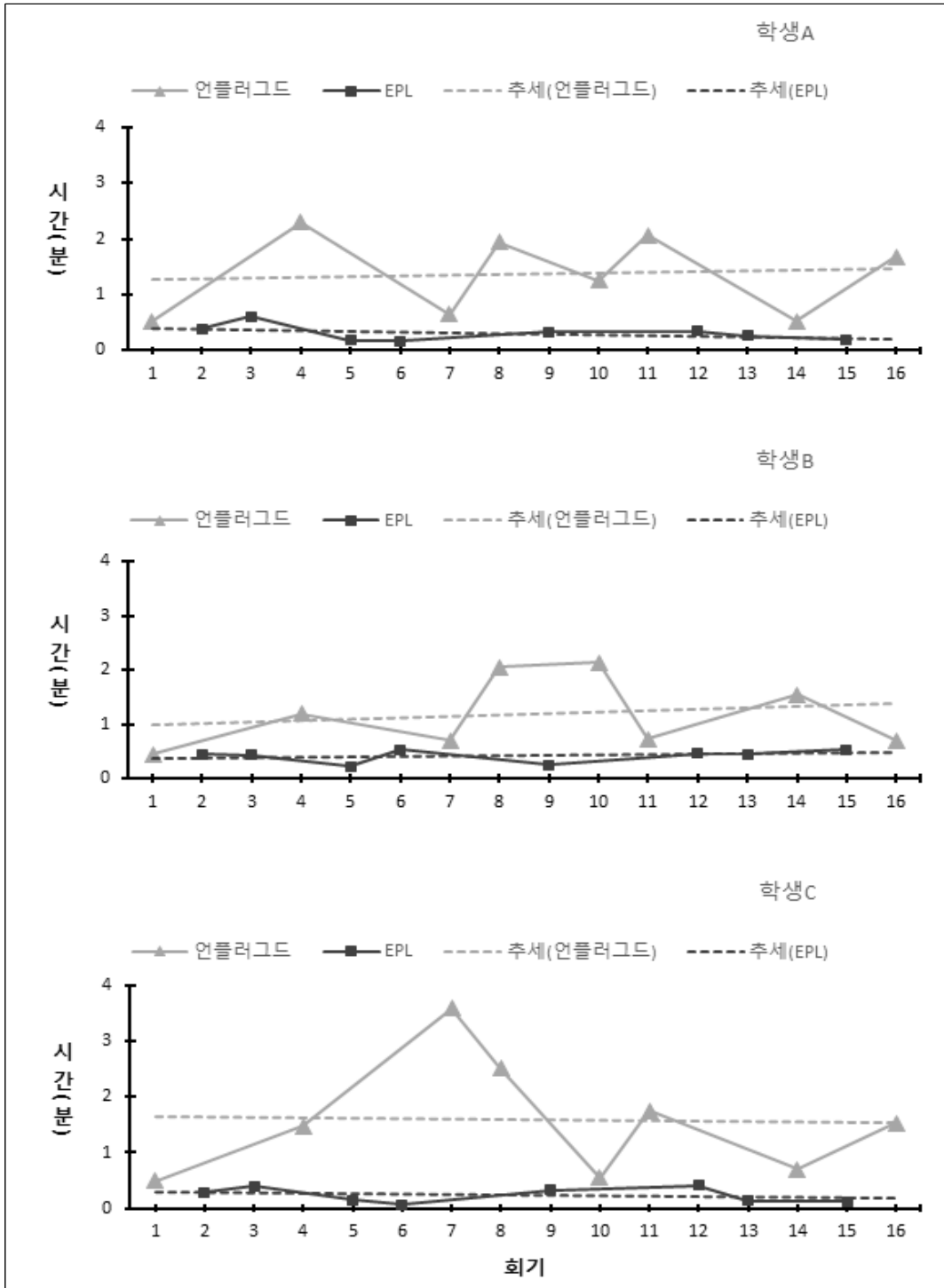
학생C의 1회 평균 구상 시간을 살펴보면 언플러그드 교육에서 1회 평균 구상 시간은 평균 1분 35초(범위 0:30~0:3:36), EPL 교육에서 1회 평균 구상 시간은 14초(범위 0:04~0:25)로 EPL 교육이 언플러그드 교육보다 평균 1분 21초 더 낮게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 2번 문제부터 시간이 급격히 상승하였으며 후반부까지 오르내리는 변화율이 크게 나타났고 경향선은 높은 수준에서 안정적인 추세를 보였다. EPL 교육의 경우 전반적으로 안정적인 경향을 유지하였으며 경향선 역시 낮은 수준에서 안정적인 추세를 보였다.

이상에서 결과를 종합해볼 때 학생 3명 모두 EPL 교육의 알고리즘 문제를 해결할 때 1회 구상 시간이 적게 나와 문제해결력의 1회 평균 구상 시간 영역에서는 EPL 교육이 효과적인 것으로 나타났다.

<표 IV-4> 수업유형별 1회 평균 구상 시간 및 평균 (단위:분)

학생	수업 유형	문제번호								평균
		1	2	3	4	5	6	7	8	
A	언플러그드 교육	0:32	2:18	0:39	1:57	1:15	2:04	0:32	1:41	1:22
	EPL 교육	0:23	0:37	0:11	0:10	0:20	0:21	0:16	0:11	0:19
B	언플러그드 교육	0:27	1:12	0:42	2:03	2:08	0:44	1:33	0:42	1:11
	EPL 교육	0:27	0:26	0:13	0:32	0:15	0:28	0:27	0:32	0:25
C	언플러그드 교육	0:30	1:29	3:36	2:31	0:33	1:45	0:42	1:32	1:35
	EPL 교육	0:17	0:24	0:09	0:04	0:20	0:25	0:08	0:07	0:14

[그림 IV-3] 1회 평균 구상 시간 그래프와 추세선



라. 오류점검 횡수 비교

오류점검이란 자신의 답이 오답임을 확인한 후 해결 과정을 복기하여 틀린 부분만을 찾아내 고치는 것을 말하며 프로그래밍의 디버깅 과정에 해당한다. 오류점검 횡수가 높을수록 알고리즘 문제해결력이 높다고 볼 수 있다. 알고리즘 문제해결을 통해 나타난 전체 학생의 각 문제별 오류점검 횡수와 그 평균은 <표 IV-5>와 같다. 회기별 평균 횡수를 살펴보면 거의 모든 문제에서 평균 1회 이하의 아주 낮은 수준으로 나타나 두 중재 모두 오류점검에 효과가 있다고 보기는 어려우며 우위적인 효과를 주장하기 어렵다. 따라서 문제해결력의 오류점검 횡수 영역에서 두 중재 조건 모두 오류점검에 미치는 효과 자체가 없었으므로 중재 간의 효과 비교는 무의미하다.

<표 IV-5> 수업유형별 오류점검 횡수 (단위: 회)

학생	수업 유형	문제번호								평균
		1	2	3	4	5	6	7	8	
A	언플러그드 교육	0	0	0	1	0	1	0	0	0.25
	PL 교육	0	0	0	0	0	0	0	0	0
B	언플러그드 교육	0	0	0	0	0	1	0	0	0.125
	PL 교육	0	0	0	0	0	0	0	0	0
C	언플러그드 교육	0	0	0	0	0	0	0	0	0
	PL 교육	0	0	0	0	0	0	0	0	0

마. 효율적인 알고리즘 구상 비교

효율적인 알고리즘 구상이란 프로그래밍 언어의 효율성의 원리와 단순화의 원리(김갑수, 2013)에 따라 반복구조를 사용해 가능한 한 자원을 적게 사용하여 논리적 구조를 간단히 표현하는 것으로서 순차적으로도 해결할 수 있지만,

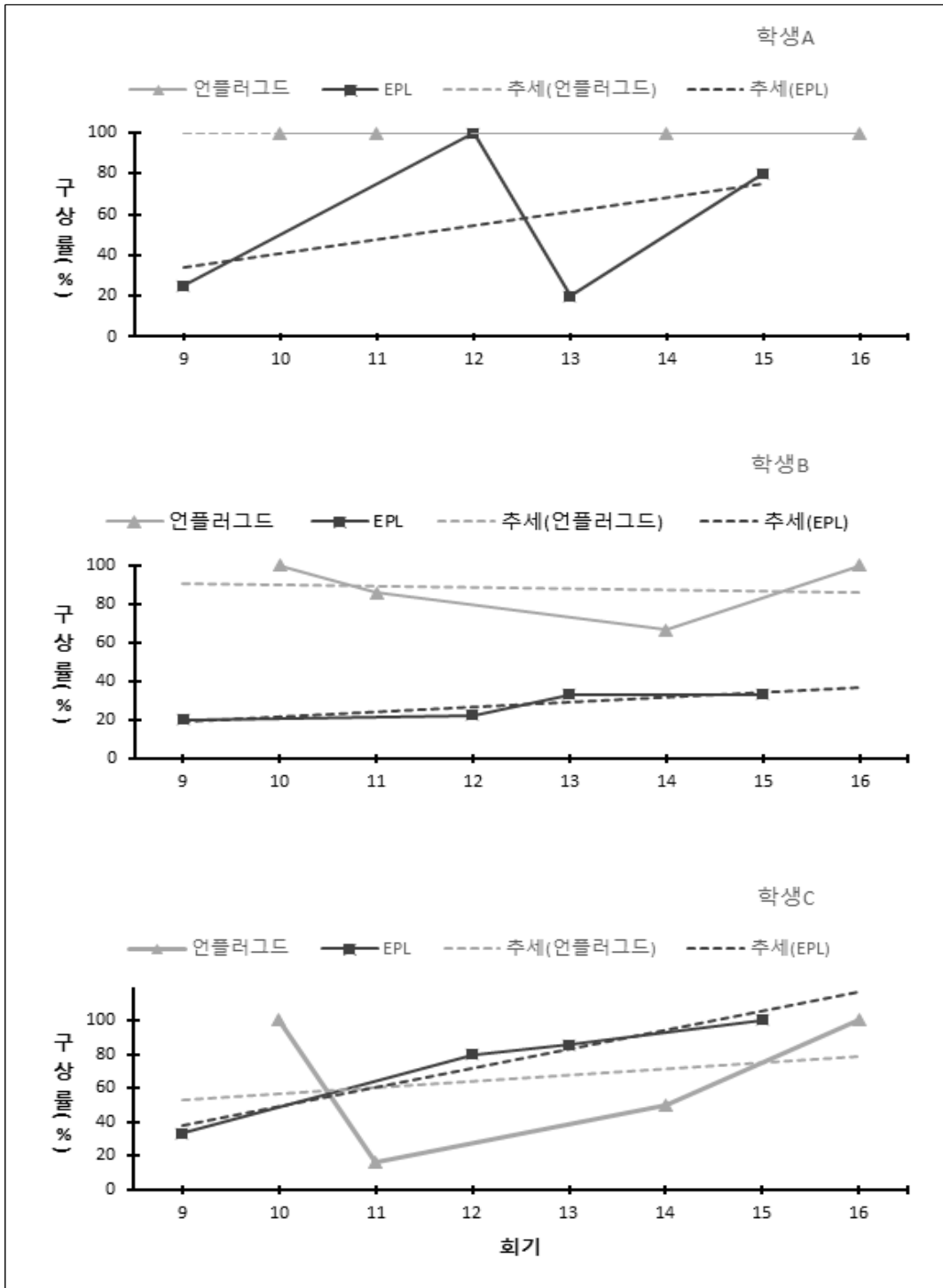
알고리즘 속에서 공통된 반복 덩어리를 찾아 묶음 짓고 반복문을 사용해 단순하고 짧게 알고리즘을 구상하는 것을 말한다. 이 연구에서는 언플러그드 교육과 EPL 교육에서 각각 5~8번 문제(9회기~16회기)에 반복문을 제시하여 순차구조와 반복구조 중 하나를 선택하여 알고리즘 문제를 해결하도록 하였으며 해결 과정에서 반복문을 구상한 비율을 비교해보고자 하였다. 따라서 반복구조를 도입한 5번 문제부터 8번 문제까지의 행동을 측정하였으며, 반복구조를 사용한 횟수를 도전 횟수로 나누어 백분율을 구하였다. 따라서 효율적인 알고리즘 구상은 백분율이 높을수록 문제해결력이 높다고 볼 수 있다. 알고리즘 문제해결을 통해 나타난 전체 학생의 문제별 효율적인 알고리즘 구상률 및 그 평균은 <표 IV-6>, 그래프와 추세선은 [그림 IV-4]와 같다.

<표 IV-6> 수업유형별 효율적인 알고리즘 구상률 및 평균 (단위: %)

학생	수업 유형	문제번호				평균
		5	6	7	8	
A	언플러그드 교육	100	100	100	100	100
	EPL 교육	25	100	20	80	56.25
B	언플러그드 교육	100	86	67	100	88
	EPL 교육	20	22	33	33	27.22
C	언플러그드 교육	100	17	50	100	67
	EPL 교육	33	80	86	100	75

학생A의 효율적인 알고리즘 구상률을 살펴보면 언플러그드 교육에서 나타난 효율적인 알고리즘 구상률은 평균 100%였으며, EPL 교육에서는 평균 56.25%(범위 20~100%)로 나타나 언플러그드 교육이 평균 43.75% 더 높게 나타났다. 그래프를 살펴보면 언플러그드 교육의 경우 모든 자료가 100%를 나타냈으며 경향선은 가장 높은 수준에서 안정적인 추세를 유지하였다. EPL 교육의 경우 전체적으로 오르내리는 변화율이 크게 나타났고 짝수 번호 문제에서 효율적인 알고리즘 구상률이 홀수 번호 문제에 비해 높게 나타났다. 경향선은 중간 수준에서 안정적인 추세를 유지하였다.

[그림 IV-4] 효율적인 알고리즘 구상률 그래프와 추세선



학생B의 효율적인 알고리즘 구상률을 살펴보면 언플러그드 교육에서 나타난 효율적인 알고리즘 구상률은 평균 88%(범위 67~100%)이며, EPL 교육에서는 평균 27.22%(범위 20~33%)로 나타나 언플러그드 교육이 평균 60.78% 더 높게 나타났다. 그래프를 살펴보면 언플러그드 교육에서는 6번 문제부터 구상률이 줄어들다가 마지막에 100%의 구상률을 보여주었으며 경향선은 우하향의 추세를 나타냈다. EPL 교육의 경우 첫 문제부터 낮은 구상률을 보여주었고 마지막 문제로 갈수록 구상률이 조금씩 상승하였다. 경향선은 낮은 수준에서 우상향의 추세를 나타냈다.

학생C의 경우 학생A와 학생B와 반대되는 결과를 나타냈다. 언플러그드 교육에서 나타난 효율적인 알고리즘 구상률은 평균 67%(범위 17~100%), EPL 교육에서는 75%(범위 30~100%)로 나타나 EPL 교육이 언플러그드 교육보다 8% 더 높게 나타났다. 그래프를 살펴보면 언플러그드 교육에서는 5번 문제에서 구상률 100%를 보이다가 그 다음 문제부터 구상률이 급격히 하강하였으며 마지막 문제에서는 다시 100%를 달성하였다. 경향성은 완만한 우상향의 추세를 보였다. EPL 교육에서는 반복구조 첫 번째 문제에서는 낮은 구상률을 보였지만 문제가 진행될수록 구상률이 상승하였고 경향선 역시 우상향의 추세를 보였다.

이상에서 결과를 종합해볼 때 학생 A와 B는 효율적인 알고리즘 구상 영역에서 언플러그드 교육이 효과적인 것으로 나타났으며, 반대로 학생C는 EPL 교육이 효과적인 것으로 나타났다.

2. 언플러그드 교육과 EPL 교육이 수업 참여행동에 미치는 효과 비교

이 연구에서는 수업유형별로 수업 참여행동을 측정하고 비교하고자 기초선, 중재, 효과적인 처치 구간을 설정해 연구를 진행하였다. 효과적인 처치 구간은 기초선과 중재를 비교하여 두 수업유형 중 효과적인 수업을 선정하였다. 수업 참여행동은 선행연구와 연구참여자의 특성을 고려하여 선정하였으며 자발적 질문하기, 주의집중하기, 과제회피 행동 3가지를 측정하였다. 전체 학생의 구간별 평균 수업 참여행동 수치는 <표 IV-7>과 같다.

수업 참여행동의 평균 점수를 살펴보면 자발적 질문은 발생률이 현저하게 낮아 두 중재 간 우위적인 효과를 주장하기 어렵다. 주의집중 행동과 과제회피

행동은 EPL 교육에서 효과적인 것으로 나타났으며 효과적인 처치에서 반복된 효과를 확인할 수 있었다.

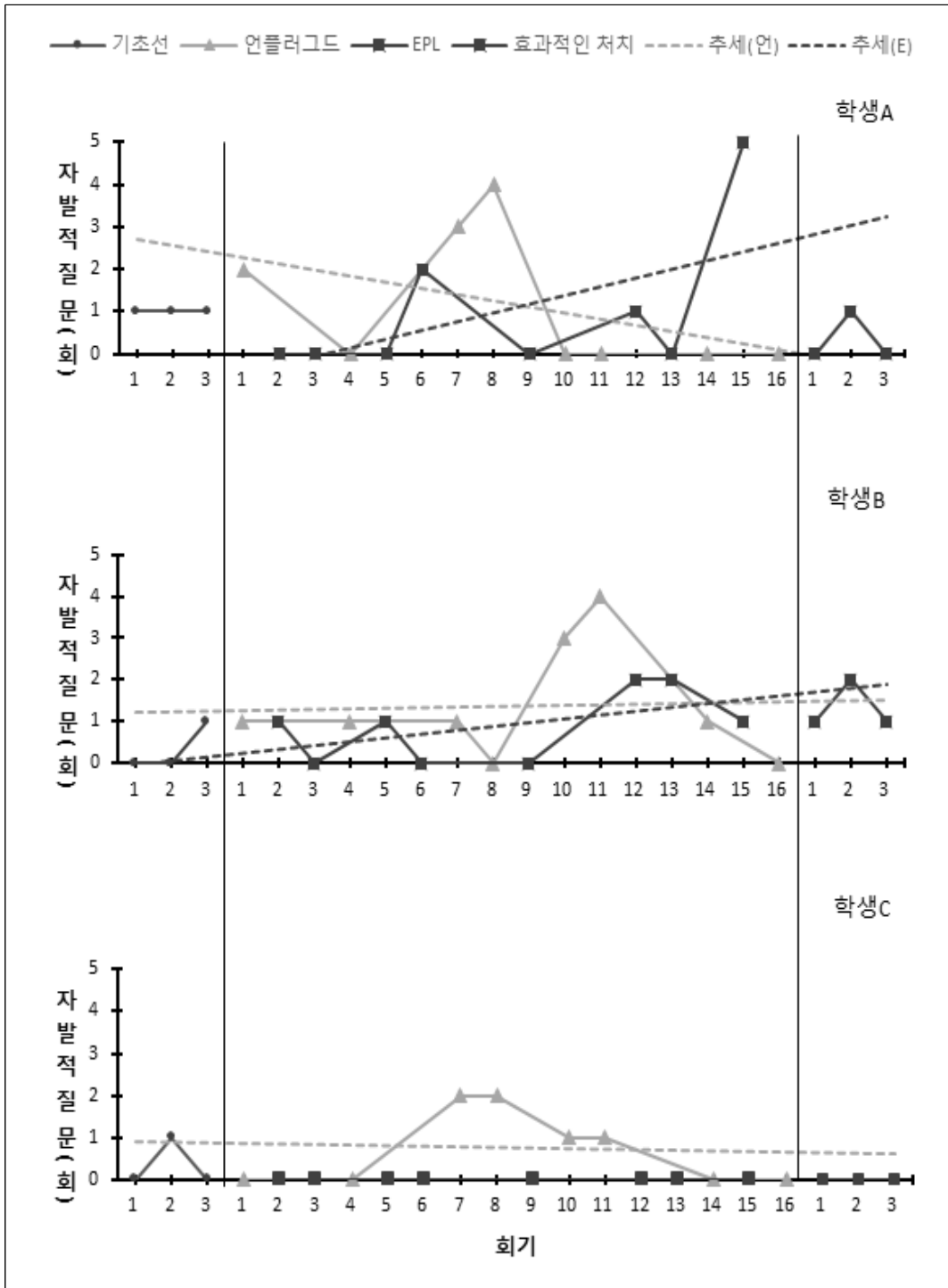
<표 IV-7> 전체 학생의 구간별 평균 수업 참여행동 수치와 범위

수업 참여행동	학생	기초선 (범위)	중재		효과적인 처치 (범위)
			언플러그드교육 (범위)	EPL 교육 (범위)	
자발적 질문	학생A	1회 (1)	1.12회 (0~4)	1회 (0~5)	0.33회 (0~1)
	학생B	0.33회 (0~1)	1.37회 (0~4)	0.87회 (0~2)	1.33회 (1~2)
	학생C	0.33회 (0~1)	0.75회 (0~2)	0회 (0)	0회 (0)
주의집중 행동	학생A	38.14% (34.44~41.11)	41.38% (24.44~58.88)	48.46% (40~61.11)	33.32% (23.33~47.77)
	학생B	28.88% (15.55~42.44)	34.71% (23.33~51.11)	47.63% (42.22~55.55)	49.25% (47.77~51.11)
	학생C	37.77% (30~44.44)	29.30% (15.55~37.77)	46.24% (36.66~52.22)	41.10% (38.88~44.44)
과제회피 행동	학생A	20.18% (17.77~25)	13.19% (8.88~17.77)	12.49% (3.33~21.11)	8.51% (6.66~11.11)
	학생B	17.77% (5.55~33.33)	17.35% (5.55~14.44)	8.60% (5.55~14.44)	9.62% (8.88~11.11)
	학생C	17.40% (5.55~28.88)	16.66% (7.77~25.55)	8.88% (4.44~13.33)	9.62% (7.77~12.22)

가. 자발적 질문 비교

자발적 질문이란 수업 상황에서 수업 내용에 대해 모르는 부분을 물어보는 행동을 말한다. 자발적 질문이 많이 나타날수록 수업 참여행동에 긍정적인 효과가 나타남을 의미한다. 이 연구에서는 자발적 질문을 측정하기 위해 빈도기록법을 통해 수업내용에 대해 모르는 부분을 물어보는 횟수를 각각 세어 측정하였다. 교사의 발화를 되물어 보는 질문(예: 교사: 풀 붙이세요./ 학생: 풀 붙여요?), 정답 확인 질문(예: 이거 맞아요?), 해결방법을 직접적으로 물어보

[그림 IV-5] 자발적 질문 그래프와 추세선



는 질문(예: 이거 어떻게 해요?), 혼잣말 질문(예: 파란색이 뭐지?)은 제외하였다. 이 연구의 전체 회기에서 나타난 전체 학생의 자발적 질문 횟수의 그래프와 추세선은 [그림 IV-5]와 같다.

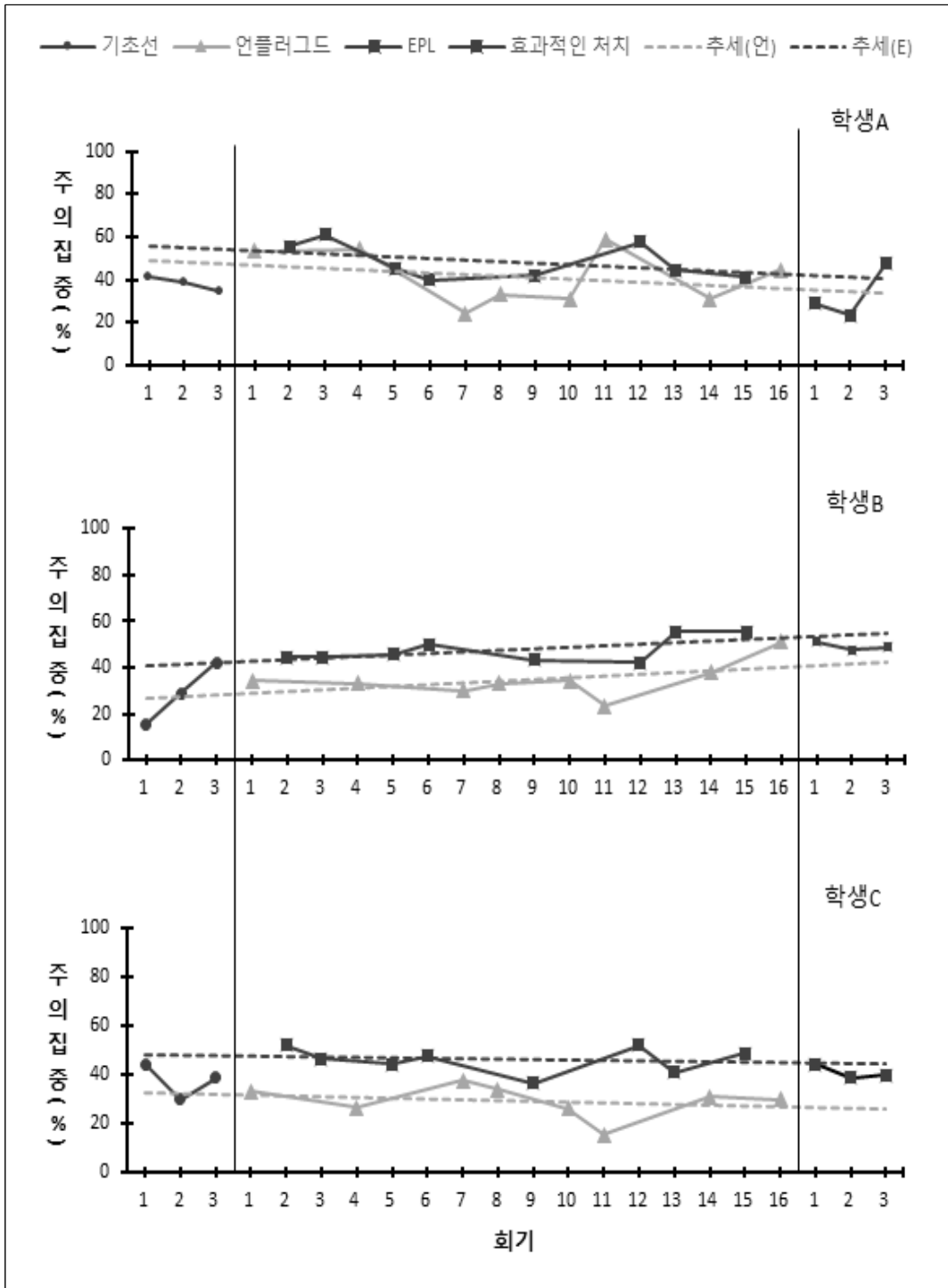
<표 IV-7>의 전체 학생의 자발적 질문의 평균 횟수나 [그림 IV-5]를 살펴보면 모든 학생은 근소한 차이로 언플러그드 교육이 높긴 하지만 기초선, 중재 구간에서 평균 2회를 밑도는 아주 낮은 수준에 머물러 있어 두 중재 모두 자발적 질문 행동에 효과가 있다고 보기는 어려우며 어느 교육이 우위적인 효과가 있다고 주장하긴 어렵다. 따라서 수업 참여행동의 자발적 질문 영역에서 두 중재 조건 모두 자발적 질문 횟수에 미치는 효과 자체가 없었으므로 중재 간의 효과 비교는 무의미하다.

나. 주의집중 행동 비교

이 연구에서 주의집중이란 발화자(교사 혹은 학생)의 시선을 5초 이상 마주치는 행동, 과제를 수행할 때 5초 이상 집중하는 행동을 말한다. 수업 중 도입-전개-정리 중 전개에 해당하는 30분을 간격기록법으로 측정하여 나타난 간격을 총 간격으로 나누어 백분율을 구하였다. 백분율이 높을수록 긍정적인 주의집중 행동이 나타난 것을 뜻한다. 수업 전개 과정 중 나타난 전체 학생의 주의집중 행동 백분율의 그래프와 추세선은 [그림 IV-6]과 같다.

학생A의 경우 <표 IV-7>의 주의집중 행동의 평균을 살펴보면 언플러그드 교육의 주의집중 행동은 평균 41.38%이며, EPL 교육은 평균 48.46%로 EPL 교육이 평균 7.08% 더 높게 나타났다. 기초선과 비교하여 평균 10.32%가 높게 나타났으며 효과적인 처치 단계에서는 기초선보다 낮게 측정되어 효과적인 처치 단계서 안정된 경향을 유지하지는 못하였다. [그림 IV-6]의 그래프를 살펴보면 언플러그드 교육의 경우 중재 초반 회기는 주의집중 행동이 기초선보다 높고 안정되게 나타났지만 중반회기부터 급격히 낮아졌으며 후반에는 오르내리는 변화율이 크게 나타났다. 경향선은 완만한 우하향의 추세를 보였다. EPL 교육의 경우 초반 회기에는 기초선보다 높고 안정되게 나타났으나 중반회기부터 다소 낮아졌지만 안정적으로 유지되었다. 경향선은 완만한 우하향의 추세를 보였다. 후속적인 처치에서 실시한 EPL 교육은 기초선보다 낮게 나타나 명확한 차이를

[그림 IV-6] 주의집중 행동 그래프와 추세선



보여주지는 않았다.

학생B의 경우 <표 IV-7>의 주의집중 행동의 평균을 살펴보면 언플러그드 교육의 주의집중 행동은 평균 34.71%이며, EPL 교육은 평균 47.63%로 EPL 교육이 12.92% 더 높게 나타났다. 기초선과 비교하여 평균 18.75%가 높게 나타났으며 효과적인 처치 단계에서는 중재보다 1.62%가 더욱 증가하여 반복된 효과를 확인할 수 있었다. [그림 IV-6]의 그래프를 살펴보면 언플러그드 교육의 경우 중재 초반 회기에서는 기초선보다 낮고 변화가 크지 않았으며 후반 회기에서 주의집중 행동이 다소 상승하였다. 경향선을 완만한 우상향의 추세를 보였다. EPL 교육의 경우 초반 회기에 기초선보다 약간 높은 주의집중 행동을 보여주었으며 중반회기 까지 큰 변화율을 보이지 않다가 후반 회기부터 주의집중 행동이 다소 상승하였다. 경향선 역시 완만한 우상향의 추세를 보였다.

학생C의 경우 <표 IV-7>의 주의집중 행동의 평균을 살펴보면 언플러그드 교육의 주의집중 행동은 평균 29.30%, EPL 교육은 46.24%로 EPL 교육이 평균 16.94%가 더 높게 나타났다. 기초선과 비교하여도 평균 8.47%로 높게 나타났으며, 효과적인 처치에서는 평균 5.14% 낮아졌지만 기초선보다 높게 나타나 반복된 효과를 확인할 수 있었다. [그림 IV-6]의 그래프를 살펴보면 언플러그드 교육의 경우 모든 회기에서 기초선보다 낮게 나타났으며 중반회기부터 주의집중이 다소 감소하였고 경향선은 완만한 우하향의 추세를 보였다. EPL 교육의 경우 초반 회기부터 기초선보다 높은 주의집중 행동을 보여주었으며 중반회기부터 약한 오르내림의 변화가 나타났고 경향선은 안정적인 추세를 유지하였다.

이상의 결과를 종합해볼 때 학생 3명 모두 EPL 교육에서 주의집중 행동이 높게 나타났으며 효과적인 처치를 통해 반복된 효과를 확인하였다. 그러나 학생A만 효과적인 처치에서 반복된 효과를 확인할 수 없었다.

다. 과제회피 행동 비교

이 연구에서 과제회피 행동이란 자리 이탈, 수업과 관련 없는 물건 만지기, 수업과 관련 없는 신체적인 행동 반복하기 등의 행동을 5초 이상 행동하는 것으로 정의하였다. 수업의 도입-전개-정리 중 전개에 해당하는 30분을 간격기록법으로 측정하여 나타난 간격을 총 간격으로 나누어 백분율을 구하였다. 백분

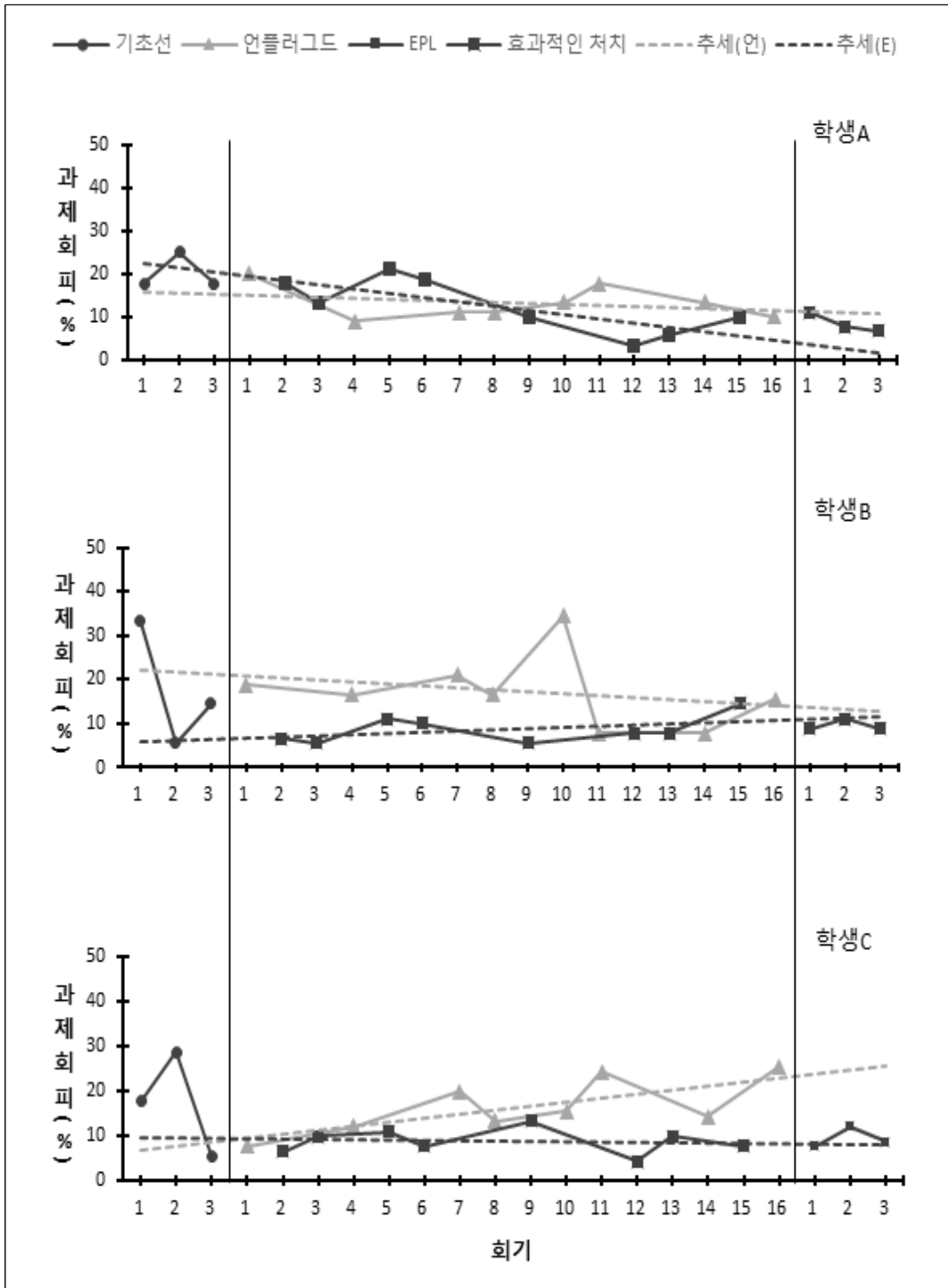
율이 낮을수록 과제회피 행동이 낮게 나타났음을 뜻한다. 수업 전개 과정 중 나타난 전체 학생의 과제회피 행동 백분율의 그래프와 추세선은 [그림 IV-7]과 같다.

학생A의 경우 <표 IV-7>의 과제회피 행동의 평균을 살펴보면 언플러그드 교육의 과제회피 행동은 평균 13.19%이며, EPL 교육은 평균 12.49%로 EPL 교육이 평균 0.7% 더 낮게 나타났다. 기초선과 비교하여 평균 7.69% 낮게 나타났으며 효과적인 처치 단계에서는 더 낮게 측정되어 반복된 효과를 확인 할 수 있었다. [그림 IV-7]의 그래프를 살펴보면 언플러그드 교육의 경우 중재 초반 회기는 과제회피 행동이 감소되고 있었지만 중반회기부터 과제회피 행동이 상승하였으나 변화폭이 크지 않게 증가하였다. 경향선은 완만한 우하향의 추세를 보였다. EPL 교육에서는 초반 회기에는 과제회피 행동이 다소 오르내림을 반복했으나 중반 회기부터 과제회피 행동이 급격히 감소하였고 경향선 역시 급격한 우하향의 추세를 보였다.

학생B의 경우 <표 IV-7>의 과제회피 행동의 평균을 살펴보면 언플러그드 교육의 과제회피 행동은 평균 17.35%이며, EPL 교육은 평균 8.6%로 EPL 교육이 평균 8.75% 더 낮게 나타났다. 기초선과 비교하여 평균 9.17% 낮게 나타났으며 효과적인 처치 단계에서는 기초선보다 낮게 측정되어 반복된 효과를 확인 할 수 있었다. [그림 IV-7]의 그래프를 살펴보면 언플러그드 교육의 경우 중재 초반 회기부터 과제회피 행동이 높고 안정되게 유지되고 있었으며 후반회기로 갈수록 과제회피 행동이 감소세가 나타났다. 경향선은 우하향의 추세가 나타났다. EPL의 경우 중재 첫 회기부터 매우 낮은 과제회피 행동을 보여주었으며 완만한 오르내림의 변화를 보여주었고 마지막 중재 회기에서 과제회피 행동이 다소 상승하여 경향선은 완만한 우상향의 추세를 보였다.

학생C의 경우 <표 IV-7>의 과제회피 행동의 평균을 살펴보면 언플러그드 교육의 과제회피 행동은 평균 16.66%이며, EPL 교육은 평균 8.88%로 EPL 교육이 평균 7.78% 더 낮게 나타났다. 기초선과 비교하여 평균 8.52% 낮게 나타났으며 효과적인 처치 단계에서는 기초선보다 낮게 측정되어 반복된 효과를 확인 할 수 있었다. [그림 IV-7]의 그래프를 살펴보면 언플러그드 교육의 경우 중재 초반 회기는 과제회피 행동이 낮게 나타났지만, 회기가 거듭될수록 과제회피 행동이 증가하여 경향선은 급격한 우상향의 추세를 보였다. EPL 교육의 경우 첫 회기부터 과제회피 행동이 낮게 나타났으며 중반 회기에 과제회피 행동이 다소

[그림 IV-7] 과제회피 행동 그래프와 추세선



상승하였지만 후반 회기에서 다시 감소하는 경향을 보여 경향선은 안정적인 추세를 유지하였다.

이상의 결과를 종합해볼 때 학생 3명 모두 EPL 교육에서 과제회피 행동이 낮게 나타났으며 효과적인 처치를 통해 반복된 효과를 확인하였다.

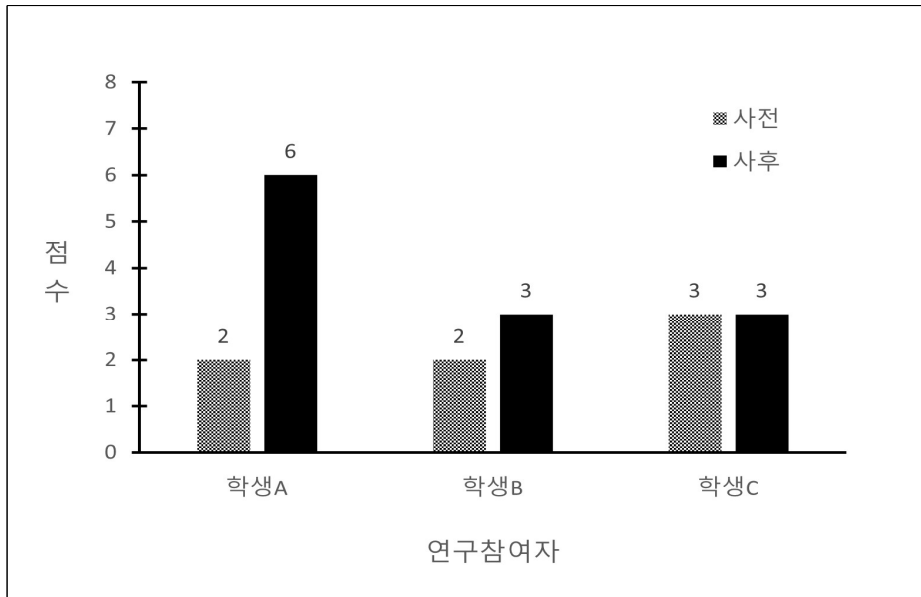
3. SW교육이 컴퓨팅 사고력에 미치는 효과

컴퓨팅 사고력은 단일한 하나의 역량으로 측정되는 것이 아니며 컴퓨팅 사고력을 구성하는 하위 역량들을 측정하여 그 효과성을 측정할 수 있다. 언플러그드 교육과 EPL 교육은 매 차시에 모든 컴퓨팅 요소를 학습하는 것이 아니라 주요 요소 1~2가지씩을 선정하여 교육하기 때문에 모든 차시가 끝나야 컴퓨팅 사고력의 주요 요소를 학습했다고 볼 수 있다. 언플러그드 교육 혹은 EPL 교육이 끝나는 시점이 컴퓨팅 사고력의 주요 요소들이 전부 학습된 상태이므로 언플러그드 교육과 EPL 교육이 장애학생의 컴퓨팅 사고력에 미치는 효과를 알아보기 위해 비버챌린지 2019년 기출문제를 중재 실시 전 1회, 중재가 끝난 후 1회 실시하였다. 컴퓨팅 사고력 평가의 총점은 8점이며, 전체 학생의 문제별 컴퓨팅 사고력 사전-사후 검사 점수는 <표 IV-8>, 학생별 컴퓨팅 사고력 사전-사후 검사 점수는 [그림 IV-8]과 같다. 학생A는 사전 검사 점수 2점, 사후 검사 점수 6점으로 4점이 상승하였다. 학생B는 사전 검사 점수 2점, 사후 검사 점수 3점으로 1점 상승하였다. 학생C는 사전 검사 점수 3점, 사후 검사 점수 3점으로 점수의 변화는 없었다.

<표 IV-8> 전체 학생의 문제별 컴퓨팅 사고력 사전-사후 검사 점수

학생	사전-사후	문제 번호								총점
		1	2	3	4	5	6	7	8	
A	사전	1	0	0	0	1	0	0	0	2
	사후	1	1	0	1	1	1	1	0	6
B	사전	0	0	0	0	1	1	0	0	2
	사후	1	0	0	1	0	0	0	1	3
C	사전	0	1	0	1	0	0	0	1	3
	사후	0	1	0	0	0	0	1	1	3

[그림 IV-8] 학생별 컴퓨팅 사고력 사전-사후 검사 점수

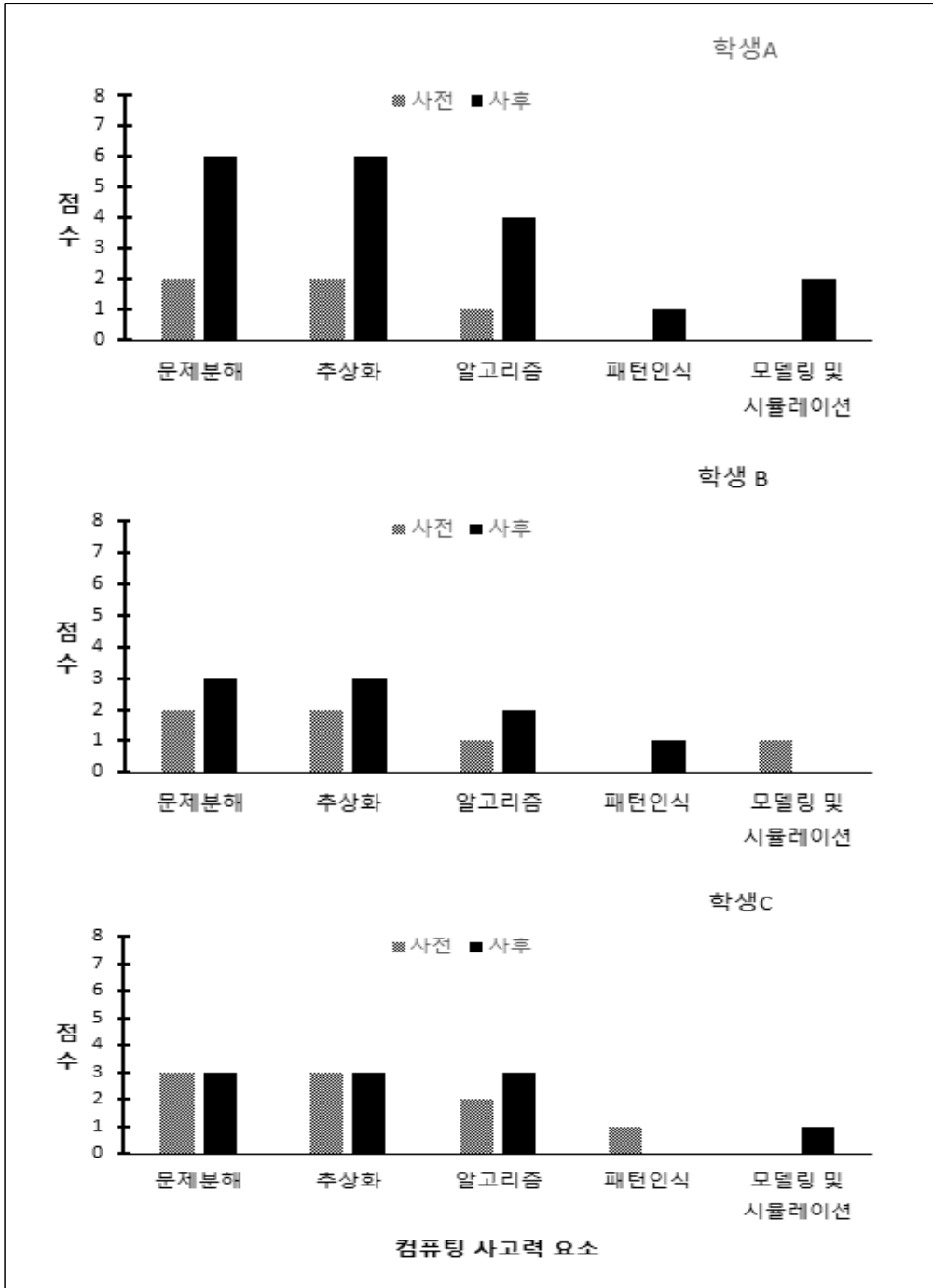


컴퓨팅 사고력 하위요소별로 살펴보면 전체 학생의 컴퓨팅 사고력 요소에 대한 사전-사후 검사 점수는 <표 IV-9>, 학생별 컴퓨팅 사고력 하위 요소를 비교한 그래프는 [그림 IV-9]와 같다.

<표 IV-9> 전체 학생의 컴퓨팅 사고력 요소 사전-사후 검사 점수

학생	사전-사후	컴퓨팅 사고력 요소				
		문제분해	추상화	알고리즘	패턴인식	모델링 및 시뮬레이션
A	사전	2	2	1	0	0
	사후	6	6	4	1	2
B	사전	2	2	1	0	1
	사후	3	3	2	1	0
C	사전	3	3	2	1	0
	사후	3	3	3	0	1

[그림 IV-9] 학생별 컴퓨팅 사고력 하위요소 비교



컴퓨팅 사고력 요소 중 3명 모두 알고리즘 요소에서 점수가 향상되었다. 학생별로 살펴보면 학생A는 문제분해, 추상화, 알고리즘, 패턴인식, 모델링 및 시뮬레이션 5가지 요소에서 모두 점수가 향상되었다. 학생B는 문제분해, 추상화, 알고리즘, 패턴인식 4가지 요소에서 점수가 향상되었으며, 모델링 및 시뮬레이션 요소만 점수가 하강하였다. 학생C는 알고리즘과 모델링 및 시뮬레이션 요소에서 점수가 향상되었다. 문제분해, 추상화 요소에서는 변화가 없었으며 패턴인식 요소는 점수가 하강하였다.

V. 논의 및 제언

이 연구는 언플러그드 교육과 EPL 교육이 지적장애 초등학생의 문제해결력, 수업 참여행동에 미치는 효과를 비교하고, 컴퓨팅 사고력에 미치는 효과를 알아보려고 하였다. 이 연구의 결론은 다음과 같다.

첫째, 측정된 종속변인을 종합했을 때 EPL 교육은 언플러그드 교육보다 지적장애 초등학생의 문제해결력과 수업 참여행동에 효과적인 것으로 나타났다.

둘째, 문제해결력을 하위 영역별로 비교하면 EPL 교육은 1회 평균 구상 시간, 정답 도달 시간, 언플러그드 교육은 도전 횟수, 효율적인 알고리즘 구성 영역에서 효과적인 것으로 나타났다.

셋째, 수업 참여행동은 EPL 교육이 언플러그드 교육보다 효과적인 것으로 나타났다.

넷째, SW교육(언플러그드 교육과 EPL 교육)을 통해 3명의 학생 중 2명의 학생에게서 컴퓨팅 사고력이 증진되었다.

본 장에서는 이와 같은 연구 결과를 토대로 이 연구의 주요한 결과의 의미와 제한점을 논의한 후 추후 연구를 위한 제언을 하고자 한다.

1. 논의

이 연구에서는 종속변인 중 알고리즘 문제를 해결하면서 나타나는 문제해결력과 수업 상황에서 보이는 수업 참여행동을 관찰 측정하여 언플러그드 교육과 EPL 교육 중 어떤 교육이 지적장애 초등학생에게 효과적인지 알아보려고 하였다. 두 종속변인 측정 결과를 종합하면 EPL 교육이 언플러그드 교육보다 지적장애 초등학생의 문제해결력과 수업 참여행동에 효과적인 것으로 나타났다. 이는 수업 상황에서 테크놀로지 기기의 사용이 학습자의 상호작용, 흥미, 주의집중을 이끌어낸다고 한 선행결과와도 일치한다(윤혁, 2016; Mize & Park, 2021).

언플러그드 교육과 EPL 교육의 가장 큰 차이점은 컴퓨팅 기기의 사용 유무이다. 이 연구에서는 EPL 교육에서 태블릿 PC를 사용하여 중재를 시행하였는데, EPL 교육은 특정 EPL 플랫폼(예: 엔트리)에 접속해야 하는 특성이 있으므로 이

연구에서는 매회기 태블릿 PC를 사용하였다. 이 때 학생들은 태블릿 PC라는 학습 도구에 대해 큰 관심과 흥미를 느꼈다. 교대중재로 인해 매회기 어떤 수업이 진행되는지 모르는 상태에서 학생들은 수업 시작 전 태블릿 PC가 있는지 없는지 확인하는 모습을 보여주었으며, 과제 수행 시에도 화면에 시선을 고정하고 주의집중을 하는 모습을 보였다. 태블릿 PC는 교실 환경에서 의미 있고 효과적인 참여와 동기부여를 촉진할 수 있는 잠재력을 가지고 있음을 보여주었다. 윤혁(2016)은 스마트 기기는 장애학생의 주의를 끄는 인터페이스를 제공하고 시각적, 청각적인 효과를 제공한다는 점에서 학생의 주의집중과 흥미를 이끌어낸다고 하였다. Mize & Park(2021)에 따르면 수업상황에서 태블릿 PC 등과 같은 테크놀로지 기기의 사용의 이점 중 하나는 수업 중 학생의 동기부여와 즐거움을 증가시킨다는 것으로 보았다. 또한, 태블릿 PC는 학문적인 동기를 부여하며, 학생들의 상호작용을 장려하고 특히 실습수업에 적극적으로 참여하게 하는 동기를 부여한다고 하였다. EPL 교육만이 가진 특성인 컴퓨팅 기기를 활용하는 교육 방식으로 인해 지적장애 초등학생의 문제해결력과 수업 참여행동에서 언플러그드 교육보다 효과가 더 높게 나타낸 것으로 보인다. 한편 한경근(2017)에 따르면 테크놀로지가 보편화되어가고 있는 현실에 비해 특수교육 분야에서는 그 논의가 활발히 이루어지지 않고 있으며, 장애학생 교육에서 테크놀로지의 활용은 특수교육 관련서비스의 보조공학 지원 수준에 머물러 있어 제4차 산업혁명 시대에 발맞춘 구체적인 논의가 이루어져야 한다고 보았다. 이 연구에서는 테크놀로지 기기인 태블릿 PC를 활용한 EPL 교육의 효과가 높게 나왔으므로 지적장애 초등학생의 SW교육에 있어서 테크놀로지 기기를 사용하는 EPL 교육이 활발하게 이루어져야 할 것임을 시사할 수 있다.

가. 언플러그드 교육과 EPL 교육이 문제해결력에 미치는 효과 비교에 대한 논의

이 연구에서는 학생들이 알고리즘 문제를 해결하며 관찰된 문제해결력이 중재별로 어떤 차이가 나는지 비교해 보고자 하였다. 연구 결과 도전 횟수와 효율적인 알고리즘 구상 능력에서는 언플러그드 교육이 효과적인 것으로 나타났으며, 1회 평균 구상 시간과 정답 도달 시간은 EPL 교육이 효과적인 것으로 나

타났다. 오류 점검 행동은 거의 나타나지 않아 유의미한 비교가 불가능하였다. 언플러그드 교육과 EPL 교육이 문제해결력에 미치는 효과 비교에 대한 주요한 논의는 다음과 같다.

첫째, 지적장애 초등학생들은 EPL 교육에서 짧은 시간 안에 문제를 해결한 것으로 나타났다. 이 연구의 지적장애 초등학생은 EPL 교육에서는 도전 횟수가 많았지만 1회 구상 시간과 정답 도달까지의 시간이 짧게 나타났다. 학생들은 오답임을 확인한 후에도 학습된 무기력이 나타나기보다 정답을 맞히기 위한 행위에 열중하여 오답 확인 직후 바로 문제에 도전하는 모습을 보여주었고 이는 짧은 시간 안에 문제를 해결하는 결과로 나타났다. EPL 교육은 명령어 블록을 차례대로 쌓는 과정을 통해 학습자들이 쉽게 알고리즘을 구성하게 하고, 각 블록들은 구문 규칙에 적합한 경우에만 결합하기 때문에 오류를 줄일 수 있어(이은경, 이영준, 2008) 학습자가 좌절을 느끼기보다 도전 의식을 갖게 하는 환경적 조건을 갖추고 있다. 또한 자신이 설계한 알고리즘을 애니메이션 효과를 통해 즉각적이고 구체적인 피드백을 주기 때문에 오류를 수정하기 쉬워 학생들이 빠른 시간 안에 문제를 해결한 원동력이 된 것으로 보인다. EPL 교육의 이러한 특징은 지적장애 학생들에게는 즉각적이고 구체적인 피드백을 제공해야 한다는 연구 결과(구슬이, 신진숙, 2018; 정해원, 2019)와도 일맥상통한다. 또한 지적장애 학생들은 잦은 실패로 인해 과제 수행을 열심히 하지 않고 과제를 빨리 포기하는 학습된 무기력을 보인다(송준만 외, 2012). 따라서 학습된 무기력이 있는 지적장애 초등학생에게 EPL 교육을 실시한다면 문제해결에 대한 두려움이 적어지고 실수를 두려워하지 않아 신속한 문제해결 수행에 긍정적인 효과를 보일 것으로 기대된다.

둘째, 연구 결과 오류점검 발생 횟수가 현저히 낮게 나타나 중재 간 비교가 불가능하였지만 이러한 결과를 통해 알 수 있는 주요 의의를 알아보고자 한다. 언플러그드 교육과 EPL 교육에서 오류점검 능력이 낮게 나온 것은 프로그래밍을 경험한 학습자들이 오류점검 활동에 어려움을 가진다는 이철현(2019)과 Vourlets et al.(2021)의 연구 결과와 일치한다. 이철현(2019)의 연구에 따르면 초등 예비교사에게 EPL 프로그래밍 학습을 실시한 후 학습에 대한 어려움을 조사했을 때 디버깅(오류 찾기, 오류 수정하기)에서 가장 높은 어려움 인식 결과를 보였다. Vourlets et al.(2021)은 연구참여자의 EPL 교육 활동 데이터를 수집하여 분석한 결과 오류점검에 대한 참가자의 숙련도 수준이 낮게 나온 것

으로 나타났다. 오류점검은 자신이 설계한 알고리즘에 대해 평가하고 수정하는 것을 말한다. 학생들은 주어진 정보와 조건을 분석하며 평가하고 오류를 수정하는 일련의 활동들을 통해 비판적 사고를 기를 수 있다(허희옥 외, 2011). 오류점검은 SW교육을 배우는 학습자에게 있어 어려움을 느낄 수 있는 요소(노지예, 이정민, 2017; 이철현, 2019)이지만 장애학생의 인지적 부하를 낮추기 위해 다양한 작업 예제를 제시하거나(시지현, 2012), 언플러그드 교육과 EPL 교육에서 오류점검 방법을 명시적으로 교수(Israel et al., 2015b)하고 오류점검의 기회와 시간을 별도로 제공하는 것이 필요할 것으로 보인다.

또한 오류점검 횟수의 빈도가 낮아 유의미한 비교는 불가능하지만, 그 안의 결과를 살펴보면 언플러그드 교육에서만 오류점검 행동이 나타났으며 EPL 교육에서는 3명 학생 모두 오류점검을 하지 않았다. 언플러그드 교육에서는 교사의 정답 확인 전까지는 자신이 설계한 알고리즘이 정답임을 확인할 수 없다. 빠른 시간 안에 알고리즘을 설계한 학생들은 교사의 정답 확인 전까지 남은 시간에 오류점검을 실행하는 모습을 보여주었다. 하지만 그 외의 문제에서는 알고리즘을 설계할 때 자신은 올바르게 코딩하고 있다는 인식을 하고 있기 때문에(이철현, 2019) 자신의 알고리즘이 오답임에도 정답이라고 착각하여 오류점검을 하지 않은 것으로 보인다. 특히 EPL 교육에서는 오류점검 행동이 한 번도 수행되지 않았다. EPL 교육은 자신이 설계한 알고리즘을 실행화면을 통해 즉각적인 시연이 이루어지고, 문법적인 오류가 발생할 때 명령 블록에 표시가 되기 때문에 오류의 위치를 찾을 수 있는 특징이 있다(오지훈 외, 2021; 이은경, 이영준, 2008). 따라서 학생들은 EPL 교육에서는 구상한 알고리즘을 처음부터 끝까지 구상하여 나열하기보다 문제를 분해한 후 앞부분의 알고리즘을 설계하고 바로 정답을 확인하였고 나머지 뒷부분은 앞부분의 정답이 확인된 후에 다시 알고리즘을 설계하는 모습을 보여주었다. 즉각적인 정답 확인은 문제를 부분으로 쪼개어 해결하는 문제 분해 능력에 도움이 되었지만, 그와 반면에 자신이 설계한 알고리즘을 평가하고 오류를 점검할 필요성을 느끼지 못하게 하여 오류점검에서 도움이 되지 못한 것으로 보인다. 이철현(2021)의 연구에서는 맞춤형 모듈식 EPL 교육 프로그램을 제작하여 프로그래밍에서 범할 수 있는 다양한 오류 예시를 제공하여 이를 수정하는 연습의 기회를 부여하였고 실험집단과 비교집단 간의 유의미한 차이를 보였다. 언플러그드 교육과 EPL 교육에서 오류점검 수행이 이루어지려면 이처럼 오류를 지닌 문제를 별도로 제시하여 오류점검 행

동을 반복하여 경험하게 한다면 SW교육 상황에서 오류를 찾고 수정하는 점검 능력에 효과적인 영향을 줄 것으로 기대된다.

셋째, 효율적인 알고리즘 구성 영역에서는 언플러그드 교육이 효과적인 것으로 나타났다. 이 연구에서는 알고리즘 문제 16문제 중 8문제는 순차구조와 반복구조 중 하나를 골라 문제를 해결할 수 있도록 하였다. 프로그래밍 언어 학습에서는 순차구조보다 반복구조를 사용하면 논리구조가 간단해지고 더 적은 자원을 사용하게 되므로(김갑수, 2013; 김세용, 2002) 순차구조를 설계하는 것보다 논리적인 사고를 필요로 한다. 언플러그드 교육에서 제시한 반복 명령어 카드는 연구참여자 모두 곱셈을 배웠기 때문에 직관적으로 알기 쉬운 X2(두번 반복하기), X3(세번 반복하기), X4(네번 반복하기)로 제작하였으며 학생들 역시 명령어 카드를 쉽게 이해하고 사용하는 모습을 보였다([그림 III-2] 참고). 따라서 학생A와 학생B는 언플러그드 교육에서 문제5~문제8까지 전부 반복 명령어 카드를 사용하는 모습을 보여주어 효율적인 알고리즘 구상률이 높게 나타났다.

반면 EPL 교육에서는 반복 블록 사용 비율이 낮게 나타났다. 이 연구에서 사용한 EPL 교육 플랫폼 엔트리에서는 반복 블록이 ‘ㄷ’ 자 모양으로 제시되어 있으며 그 안에 블록을 하나 혹은 여러 개를 넣을 수 있고 횟수를 따로 설정해 주어야 한다. ‘ㄷ’ 자 모양의 반복 블록은 그 안에 블록을 넣을수록 크기가 세로로 늘어나지만 기본 제시 모양은 한 개의 블록만 넣을 수 있을 정도의 크기로 제시되어 학생들이 직관적으로 사용하기 어려워하는 모습을 보였다([그림 III-2] 참고). 반복 블록 명령어의 처음 도입 시에는 교수자의 설명이 있었으므로 어려움 없이 문제를 해결하였으나 반복 블록 명령어가 두 번째로 제시된 12 회기(EPL 교육)에서는 반복 블록 명령어의 설명 없이 해결해야 하고 블록 모양이 직관적으로 이해하기 쉽지 않아 도전 횟수나 정답까지 걸린 시간이 늘어났음을 알 수 있었다. 지적장애 학생들은 주로 전조작기나 구체적 조작기에 머물러 있어(송준만 외, 2012) 형태나 위치를 변형시켜도 항상 동일함을 인지하는 보존 개념이 발달하지 못하여 반복 블록 명령어를 이해하지 못한 것으로 보인다. 미국의 대표적 EPL 사이트인 스크래치를 활용한 교육 사례에서도 초등학교 4~5학년 학생들을 대상으로 EPL 교육을 하였는데 학습자들은 반복 블록 그림을 잘 이해하지 못하여 반복 블록을 조립하는 데 문제를 나타냈다(Franklin et a

1., 2020). 코딩 문법이 어려워질수록 인지적 과부하로 학습자는 흥미를 잃게 된다. 지적장애 학생들이 SW교육에 흥미를 지속적으로 가질 수 있도록 언플러그드 교육을 통해 반복 구조에 대한 개념을 알기 쉽게 교수한 후, EPL 교육에서 반복 학습을 제공한다면 반복문을 사용한 효율적인 알고리즘을 설계할 것으로 기대된다.

나. 언플러그드 교육과 EPL 교육이 수업 참여행동에 미치는 효과 비교에 대한 논의

이 연구에서는 언플러그드 교육과 EPL 교육을 실시하여 지적장애 초등학생의 수업 참여행동(자발적 질문하기, 주의집중 행동, 과제회피 행동)에서 나타나는 영향을 기초선, 중재, 효과적인 처치를 통해 비교해보고자 하였다. 이 연구에서는 수업 참여행동 중 주의집중 행동과 과제회피 행동에서 EPL 교육이 효과가 긍정적으로 나타나 효과적인 처치는 EPL 교육으로 시행하였다. 수업 참여행동에 대한 주요한 논의는 다음과 같다.

첫째, EPL 교육은 언플러그드 교육보다 지적장애 초등학생의 주의집중 행동 증가에 효과적인 것으로 나타났다. 이와 같은 결과는 장애학생을 대상으로 EPL 교육을 했을 때 주의집중 시간이 향상된 결과를 얻은 정소영(2019)의 연구 결과와도 일치한다. EPL 교육은 명확한 목표를 제시하여 프로그래밍 과제를 수행하며, 스크립트 작성에 대한 구체적인 피드백을 주고, 쉽고 직관적인 인터페이스 환경으로 도전과 기술이 조화되는 특성을 가져 학생들의 몰입을 경험(이은경, 이영준, 2008)시키기 때문에 교수 내용에 더 집중할 수 있었던 것으로 보인다. 이 연구의 지적장애 초등학생들은 EPL 교육에 대한 수용적인 태도를 보여주었으며 자신이 세운 알고리즘이 틀렸을 때 부끄러워하거나 불안해하지 않고 문제를 해결하려는 모습을 보여주었고, 이를 통해 인내심이 길러져 주의집중에 효과적인 것으로 보인다. 따라서 EPL 교육은 지적장애 초등학생의 주의집중에 긍정적인 영향을 끼치며 동기부여와 수업 참여행동을 높일 중재방법으로 교육 현장에서 적용될 수 있을 것으로 기대된다.

둘째, EPL 교육은 언플러그드 교육보다 과제회피 행동 감소에 효과적인 것으로 나타났다. 이는 학습자가 배움의 주체가 되는 능동적인 학습이 학생의 수업

방해 행동이 감소된다는 연구결과와 일치하는 결과이다(김경신, 김미경, 2012; 오영범, 2017; Settles, 2009). 능동적인 학습이란 학생들이 수업에 참여할 기회를 주어 수업에 반응하게 하는 것이다(안성우, 서유경, 2019). EPL 교육은 학생들이 직접 알고리즘을 설계하고 개개인이 독립적으로 애니메이션 효과를 통해 즉각적인 피드백을 받는 능동적인 학습 환경을 제공한다. 능동적인 학습은 시간이 오래 걸리거나 비용이 많이 드는 현대 테크놀러지 학습 상황에서 동기를 유발하는 효과가 있다(Settles, 2009). 또한, 수업에 대한 만족도, 자신감, 흥미도가 높을수록 수업 태도가 향상(김경신, 김미경, 2012)되는데 학습자가 배움의 주체가 되는 EPL 교육은 학습자 자신이 능동적인 참여자와 평가자가 되는 경험을 제공하여 지적장애 초등학생의 과제회피 행동이 감소하는 교육적 효과가 나타난 것으로 보인다.

셋째, 자발적 질문 횟수는 두 중재 모두 발생 횟수가 현저하게 낮게 나타나 중재 간 비교가 불가능하였지만 측정된 결과를 통해 알 수 있는 주요 의의를 알아보려고 한다. 측정된 결과를 살펴보면 자발적 질문 행동은 언플러그드 교육에서 더 많이 나타난 것을 알 수 있었다. 언플러그드 교육은 만들기 활동, 알고리즘 학습을 위한 조작 활동, 낚시 놀이 등의 수업으로 이루어져 규칙에 관한 질문이나 수업 내용을 파악하기 위한 자발적 질문들이 이루어졌다. 반면 EPL 교육의 경우 자신이 설계한 알고리즘은 애니메이션 효과를 통해 즉각적이고 생동감 있는 시연이 이루어진다. 또한, 오류 발생 시 즉각적이고 구체적인 피드백 전달이 가능하여(이은경, 이영준, 2008) 교사에게 질문하지 않고 독립적으로 해결하는 모습을 보였다. 질문이란 인지 수준을 확장시키며(김미영, 김은경, 2012) 주체적인 학습을 가능하게 하는 중요한 역할(양미경, 2007)을 한다. 따라서 EPL 수업에서도 배운 내용에 대한 지식의 재구조화가 일어날 수 있도록 직접교수 방식으로 지도하기보다 탐구 수업 형식이나 거꾸로 교실(flipped learning)의 방법(김명숙, 2015)으로 수업을 설계한다면 EPL 교육에서도 자발적 질문이 많이 나올 것으로 기대된다.

다. SW교육이 컴퓨팅 사고력에 미치는 효과에 대한 논의

이 연구에서는 언플러그드 교육과 EPL 교육, 즉 SW교육이 지적장애 초등학생

의 컴퓨팅 사고력에 영향을 미치는지 알아보았다. 연구참여자 3명 중 2명의 학생이 SW교육을 통해 컴퓨팅 사고력 점수가 향상되었다. 이는 비장애학생들에게 SW교육을 적용한 결과 컴퓨팅 사고력에 효과가 있다는 선행 연구의 결과와 일치하며(강은희, 2019; 강은희, 2018; 김준호, 2022; 나청수 외, 2018; 노지예, 이정민, 2018; 박선주, 2019; 송옥지 외, 2020; 이철현, 2017; 조은진, 2022), 장애학생들을 대상으로 프로그래밍 교육을 실시하여 컴퓨팅 사고력에 효과적임을 밝힌 Munoz et al.(2018)의 연구 결과와도 일치함을 알 수 있다. 언플러그드 교육은 컴퓨터 과학의 기본 개념을 통해 추상화의 개념을 익히고 문제해결 초기 능력이 향상된다(Bouck & Yadav, 2022; Hufad et al., 2021). 또한, EPL 교육은 컴퓨터 소프트웨어가 애니메이션으로 표현되어 학생들의 상상력을 증가시키고 문제에 대한 해결책을 더 쉽게 찾도록 도움을 주며, 다양한 프로그래밍 콘텐츠 조작과 프로그래밍 활동 참여는 컴퓨팅 사고력과 관련된 기술과 이해 영역을 다양한 방식으로 신장할 수 있다(Fagerlund et al., 2021; Voskoglou & Buckley, 2012). 이처럼 언플러그드 교육과 EPL 교육, 즉 SW교육은 문제해결 능력의 향상과 컴퓨팅 사고력의 신장을 가져오는 교육 방법이기 때문에 컴퓨팅 사고력에 긍정적인 영향을 미친 것으로 보인다. 컴퓨팅 사고력은 논리, 판단, 분석, 종합적인 능력이 주도하는 비판적 사고이다(Voskoglou & Buckley, 2012). 따라서 SW교육은 장애학생의 비판적 사고력과 같은 고차원적 사고력을 습득하고 신장시키기 위한 중재 방법으로서 교육 현장에 적용될 수 있을 것으로 기대된다.

반면 학생C의 경우 컴퓨팅 사고력 검사에서 사전-사후 검사 점수의 변화는 없었다. 이는 SW교육이 컴퓨팅 사고력에 효과를 나타냈다는 앞서 제시한 선행 연구와는 상반되는 결과이다. 이와 같은 결과는 이 연구의 중재 기간 혹은 평가 방법의 요인으로 인한 결과로 보인다. 비버챌린지 검사는 전이된 컴퓨팅 사고력을 측정하는 특징(노지예, 이정민, 2018)을 가지며, Ayer(2018) 따르면 컴퓨팅 사고력은 장기간에 걸쳐 발전하는 능력으로 보였다. 따라서 학생C의 경우는 16차시의 단기간의 교육만으로 컴퓨팅 사고력을 향상시키기 어려웠던 것으로 보인다. 컴퓨팅 사고력의 신장을 위해서는 SW교육이 단기간으로 끝내는 것이 아니라 타 교과와 융합교육을 통해 장기적으로 이루어진다면 지적장애 초등학생의 컴퓨팅 사고력의 효과를 정확히 측정할 수 있을 것으로 사료된다. 또한 유인환(2018)의 연구에서는 컴퓨팅 사고력의 평가를 학생의 개발일지와 학생

관찰을 통해 확인하였는데 향후 연구에서는 지적장애 초등학생에게 나타난 컴퓨팅 사고력의 변화를 측정하기 위해 컴퓨팅 사고력 검사뿐만 아니라 학생 관찰, 산출물 평가, 학습자 인터뷰 등 지적장애 학생의 특성에 적합한 추가적인 평가 방법의 모색이 필요할 것으로 보인다.

컴퓨팅 사고력의 하위 요소별로 살펴보면 컴퓨팅 사고력 검사 결과 3명 학생 모두 알고리즘 요소가 향상되었음을 알 수 있었다. 이는 언플러그드 교육과 EPL 교육 모두에서 알고리즘 학습을 진행하였는데, 알고리즘에 대한 직접적인 학습으로 알고리즘 영역에서 모두 점수가 향상된 것으로 보인다. 이는 언플러그드 교육과 EPL 교육이 알고리즘적 사고, 추상화, 분해 능력을 향상시키고 개발된다고 본 Moschella(2019)의 발언을 뒷받침한다고 볼 수 있다. 특히 알고리즘은 수학이나 과학 분야에서 문제해결을 위해 사용되는 능력이며 창의적 문제해결력을 길러주는 중요한 학습능력인데(한옥영, 김재현, 2011), SW교육은 장애 학생의 컴퓨팅 사고력의 알고리즘 능력을 신장시키는 데 효과적인 중재로 활용될 수 있을 것으로 기대된다.

학생B의 경우 컴퓨팅 사고력 하위 요소에서는 모델링 및 시뮬레이션 요소를 제외한 4가지 요소가 상승한 것으로 나타났다. 모델링 및 시뮬레이션 요소가 신장하지 못한 것은 지적장애 학생이 가진 외부지향성(outerdirectedness)에 대한 특성 때문으로 보인다. 지적장애 학생은 문제를 해결할 때 자신의 내적 인지적 능력을 사용하기보다 외부에서 단서를 얻으려고 하는 외부지향성의 특성을 보인다(송준만 외, 2012). 학생B는 외부의 단서에 기대는 경우가 많았고 스스로 답의 경로를 추적하는 횟수나 시간이 적었기 때문에 정보 경로를 추적함으로써 발달되는 모델링 및 시뮬레이션 요소의 점수가 하강한 것으로 보인다. SW교육에서는 지적장애 초등학생의 특성을 고려하여 인지적 도전 과제를 부여하되 적절한 스캐폴딩을 제공(Pakarinen et al., 2011)하여 학생들이 SW교육에 높은 참여도가 나타날 수 있도록 해야 할 것이다.

2. 연구의 제한점

이 연구의 제한점은 다음과 같다.

첫째, 이 연구는 3명의 지적장애 초등학생을 대상으로 실시한 단일대상연구

이며 무선 표집이 아닌 의도적인 표집을 했기 때문에 연구 결과를 일반화하는 데 어려움이 있다. 따라서 추후 연구에서는 다른 학년 군이나 다양한 학생들을 대상으로 연구를 시도해 볼 필요가 있다.

둘째, 이 연구의 컴퓨팅 사고력은 3명 중 2명의 학생에게만 효과적으로 나타났다. SW교육을 통해 컴퓨팅 사고력이 유의하게 높아졌다는 연구가 많이 있으므로 이 연구와 다른 컴퓨팅 사고력 측정 방법을 이용해 장애학생의 컴퓨팅 사고력을 측정하는 연구를 시도해 볼 필요가 있다.

셋째, 연구 결과가 호손 효과(hawthorne effect)에 의한 것일 수도 있어 연구 결과를 일반화하는 데 어려움이 있다. 핸드폰을 연구참여자 앞에 두고 영상을 찍었기 때문에 수업 시작 전에 카메라에 대고 손으로 브이자를 하거나 수업 중 교사를 보지 않고 카메라를 쳐다보는 행동을 한 학생들이 있었다. 카메라를 통해 연구참여자가 실험상황이라는 것을 인식했기 때문에 추후 다른 연구에서는 호손 효과가 발생하지 않게 실험상황을 만들어 연구를 진행할 필요가 있다.

넷째, 수업 참여행동의 연구 결과가 검사자나 검사장소에 의한 것이 수도 있다. 검사자는 연구참여자의 특수학급 담임교사이며 검사장소는 특수학급이었다. 교사와 래포형성이 된 상태였기 때문에 교사에게 질문할 때 어려움 없이 질문했을 수 있으며, 주의집중 행동에서는 학기 초에 수립했던 학급 규칙들(예: 주의집중이 흐트러질 때 주의집중 구호 외치기 등)이 그대로 적용되었기 때문에 이에 따른 수업 참여행동 결과에 영향을 끼쳤을 수 있다.

다섯째, 이 연구는 약 12주간 언플러그드 교육과 EPL 교육을 실시하여 지적장애 초등학생의 문제해결력, 수업 참여행동을 비교하였으며 수업 참여행동에 서만 기초선, 중재, 효과적인 처치 구간으로 그 효과를 알아보았으며 일반화와 유지 단계는 적용하지 못하였다. 추후 연구에서는 수업 참여행동 뿐만 아니라 문제해결력 영역에서도 기초선과 효과적인 처치 구간을 도입하고, 장기적인 연구를 통해 일반화와 유지 단계까지 적용하여 두 가지 중재의 효과를 비교할 필요가 있다.

3. 제언

후속 연구를 위한 제언을 제시하면 다음과 같다.

첫째, 텐저블 코딩 교육이나 피지컬 컴퓨팅 교육 연구도 필요하다. EPL 교육의 경우 컴퓨터나 태블릿 PC를 이용해야 하므로 기본적인 컴퓨터 소양 교육이 필요하다. 또한 알맞은 블록을 선택하기 위해서는 글자를 읽을 수 있는 능력이 필요하다. 따라서 글자를 모르는 학생이나 마우스 혹은 손가락 조작이 어려운 중증의 장애학생들에게 적용하기에 한계가 있다. 텐저블 코딩은 물리적 조작 도구를 이용하여 직관적으로 알아볼 수 있는 그림으로 제작되는 경우가 많아 SW교육을 처음 접하는 장애학생들에게 쉽게 가르칠 수 있을 것으로 예상된다. 피지컬 컴퓨팅 교육은 로봇이나 센서보드, 드론 등을 이용해 자신이 설계한 알고리즘을 실제 사물의 움직임을 통해 확인할 수 있으므로 장애학생의 수업 참여행동이나 흥미도를 높일 수 있을 것이다.

둘째, 장애학생의 컴퓨팅 사고력을 측정할 수 있는 도구 개발 연구가 필요하다. 이 연구의 컴퓨팅 사고력 측정 도구인 비버챌린지 문제는 연령별 혹은 학년별로 난이도를 달리하여 제시하기 때문에 제일 낮은 연령의 문제를 사용하였다. 문제들은 비장애학생 기준으로 만들어진 문제이며 그림과 줄글로 제시되어 있어 글을 읽고 내용을 파악할 줄 아는 능력이 선행되어야 한다. 따라서 장애영역별 혹은 학습 수준별로 장애학생의 컴퓨팅 사고력을 정확히 측정할 수 있는 타당화된 도구에 관한 연구가 필요하다.

셋째, 지적장애 초등학생뿐만 아니라 다른 장애 영역, 다른 학년 군 학생을 대상으로 연구를 시행하여 SW교육의 효과를 검증해 볼 필요가 있다. 학년이 높아질수록 SW교육 시수는 늘어나기 때문에 초등학생뿐만 아니라 중·고등학교 장애학생을 위한 SW교육 연구도 이루어져야 할 것이다.

넷째, 학습모형을 적용한 SW교육 혹은 교과와 연계한 SW교육 연구도 필요하다. SW교육 학습모형에는 컴퓨팅 사고력 기반 창의적 문제해결(CT-CPS)모형, 언플러그드 활동 모형, 언플러그드 협동학습 모형, 소프트웨어 활용 수업 모형 등(채성은, 허유성, 2021)이 있다. SW교육 수업모형을 적용해 수업을 시행하여 연구결과를 검증해 볼 필요가 있다. 교육부(2015a)의 SW교육 운영지침에서는 융합교육을 위해 교과 활동과 SW교육을 연계할 것을 권장하고 있다. 장애학생에게도 창의·융합교육이 제공될 수 있도록 SW교육과 교과 연계 교육에 대한 연구가 이루어져야 할 것이다.

참고문헌

- 강두봉 (2019). 시맨틱 네트워크 분석을 이용한 국내외 언플러그드 활동 관련 연구 비교. **컴퓨터교육학회논문지**, 22(4), 21-34.
- 강옥려, 고승희 (2005). 인지-메타인지전략 훈련이 학습장애아동의 수학문장제 문제해결력과 자기효능감에 미치는 효과. **특수교육저널: 이론과 실천**, 6(3), 135-154.
- 강윤아, 신현기 (2019). 초등학교 교사의 통합학급 운영 경험에 대한 자문화기술지. **통합교육연구**, 14(1), 1-28.
- 강은희 (2019). 앱 인벤터를 활용한 SW교육이 컴퓨팅 사고력과 창의적 문제해결력에 미치는 영향. 미간행 석사학위논문, 한밭대학교 대학원, 대전.
- 강은희 (2018). 컴퓨팅 사고력을 기르기 위한 비버챗린지의 효과에 관한 연구. 미간행 석사학위논문, 경상대학교 교육대학원, 진주.
- 고정현, 이한얼 (2015). 오르프 교수법을 적용한 음악수업이 통합학급 지적장애 학생의 수업참여도와 교육관계에 미치는 영향. **지적장애연구**, 17(1), 113-132.
- 곽은정, 신진숙 (2011). 협력교수가 초등학교 지적장애아동의 문제행동과 수업참여행동 및 교사와 일반아동의 인식에 미치는 영향. **지적장애연구**, 13(3), 69-90.
- 교육부 (2015a). **소프트웨어 교육 운영 지침**. 서울: 교육부.
- 교육부 (2015b). **교육부 고시 제2015-81호 [별책 3] 기본 교육과정**. 서울: 교육부.
- 교육부 (2017a). **2017년 소프트웨어(SW)교육 선도교원 연수 교재(초등)**. 대구: 한국교육학술정보원.
- 교육부 (2017b). **언플러그드를 활용한 소프트웨어(SW)교육(초등)**. 대구: 한국교육학술정보원.
- 교육부 (2018). **2018년 소프트웨어(SW)교육 선도교원 연수 교재(초등)**. 대구: 한국교육학술정보원.
- 교육부 (2021). **더 나은 미래, 모두를 위한 교육 2022 개정 교육과정 총론 주요 사항(시안)**. 서울: 교육부.

- 교육부 (2022a). **교육부 고시 제2022-33호 [별책 1] 초·중등학교 교육과정 총론**. 서울: 교육부.
- 교육부 (2022b). **교육부 고시 제2022-34호 [별책 3] 기본 교육과정**. 서울: 교육부.
- 구슬이, 신진숙 (2018). 직접교수에 대한 수행피드백이 특수교사의 교수 충실도에 미치는 효과: 중등도 지적장애 초등학생의 덧셈 기술 습득을 중심으로. **특수아동교육연구**, 20(1), 143-172.
- 구슬이, 신진숙 (2019). 수정된 도식기반전략교수가 중등도 지적장애 초등학생의 뺄셈 문장제 문제 해결력에 미치는 효과. **특수아동교육연구**, 21(3), 141-166.
- 구영은 (2015). 초등학교 저학년 학생을 대상으로 한 놀이학습 기반 언플러그드 교육의 효과성 분석. 미간행 석사학위논문, 경인교육대학교 대학원, 인천.
- 구은정 (2016). 로봇 프로그래밍 교육이 학습장애학생의 학습된 무기력과 자발적 언어 사용에 미치는 영향. **한국게임학회논문지**, 16(1), 93-102.
- 국립특수교육원 (2019). **소프트웨어(SW)와 친구하자 지도서**. 아산: 국립특수교육원.
- 국립특수교육원 (2022). **2022특수교육통계**. 아산: 국립특수교육원.
- 권숙진 (2021). 유아교육에서 텐저블 미디어를 활용한 디지털 놀이 개념 탐색 연구: 컴퓨팅 사고력 향상을 중심으로. **교육종합연구**, 19(1), 1-15.
- 권순황 (2018). 특수교육 교육과정 적용 대상학생의 소프트웨어 교육에 관한 분석. **교육혁신연구**, 28(4), 441-460.
- 권은정 (2008). 놀이를 통한 알고리즘 개념 학습이 학습 동기 및 학업 성취도에 미치는 영향. 미간행 석사학위논문, 한국교원대학교 교육대학원, 충주.
- 권정인 (2014). Computational Thinking 기반의 교수-학습이 학습자의 창의적 문제해결에 미치는 효과성 연구. 미간행 박사학위논문, 성균관대학교 대학원, 서울.
- 권효진 (2012). 보편적 학습설계를 적용한 과학 수업이 중학교 장애학생과 비장애학생의 과학학업성취도, 수업참여도 및 교수·학습활동에 미치는 영향. 미간행 박사학위논문, 이화여자대학교 대학원, 서울.
- 길한아, 손승현 (2018). 마인드맵을 활용한 읽기·쓰기 교수방법이 자폐스펙트럼 장애 청소년의 질문에 대한 반응과 작문의 길이에 미치는 효과. **특수교육저**

널: 이론과 실천, 19(2), 49-74.

- 김갑수 (2010). 초등학생들의 창의력과 논리력 향상을 위한 프로그래밍 언어 교수전략에 관한 연구. **정보교육학회논문지**, 14(1), 89-97.
- 김갑수 (2013). 정보영재아를 위한 개념 클립(CLIP)을 기반 프로그래밍 교육 방법 연구. **한국초등교육**, 24(1), 291-310.
- 김거현, 유인환 (2017). 앱 인벤터 활용 SW 교육이 초등학생이 컴퓨팅 사고력과 컴퓨터에 대한 태도에 미치는 영향. **정보교육학회논문지**, 21(4), 371-380.
- 김경신, 김미경 (2012). 북아트를 활용한 기본교육과정 국어과 지도가 장애학생들의 국어과에 대한 태도에 미치는 효과. **정서·행동장애연구**, 28(3), 125-145.
- 김경희 (2015). 동기화 조작(E0)과 에코익 촉구가 자폐스펙트럼 장애아동의 ‘의문사로 질문하기’ 형성에 미치는 효과. 미간행 석사학위논문, 공주대학교 특수교육대학원, 공주.
- 김남희 (2011). 가족이 참여한 자기결정교수학습모델(SDLMI) 적용이 문제행동을 보이는 중학교 특수학급 학생의 수업참여행동 및 국어과 교육목표 성취도에 미치는 영향. 미간행 박사학위논문, 이화여자대학교 대학원, 서울.
- 김돈정 (2016). 초등학교 돌봄교실 환경에서 로봇기반 프로그래밍 학습효과의 지역 및 성별에 따른 차이 분석 - 창의성, 문제해결력, 몰입도를 중심으로. 미간행 박사학위논문, 건국대학교 대학원, 서울.
- 김동선 (2020). 정보통신활용 교과에 도입된 소프트웨어 교육에 대한 특수교사의 인식 조사. 미간행 석사학위논문, 한국교원대학교 대학원, 청주.
- 김동일, 손지영, 김경선, 김주연, 김효정 (2012). 장애대학생 대상 IT 교육 프로그램에 대한 학습경험의 분석. **특수교육재활과학연구**, 51(1), 287-312.
- 김명숙 (2015). 교실수업에서 ‘질문하는 일’의 의미. **초등교육연구**, 28(3), 1-19.
- 김미영, 김은경 (2012). 동기절차와 결합한 질문하기 중재가 자폐스펙트럼장애 아동의 자발적 질문하기와 어휘습득에 미치는 영향. **자폐성장애연구**, 12(3), 1-21.
- 김미화, 최윤희, 박나리 (2020). 특별한 관심을 반영한 시각적 지원이 통합된 자폐성 장애유아의 참여행동에 미치는 영향. **유아특수교육연구**, 20(1), 1-19.
- 김민정, 이원규, 김자미 (2017). 컴퓨팅사고력 측정에 사용되고 있는 도구 분석

- 을 통한 새로운 검사도구 개발방향 제시. **컴퓨터교육학회논문지**, 20(6), 17-25.
- 김병곤 (2019). 칸 아카데미 수학학습 프로그램이 지적장애 학생의 수학 연산 학습 성취효과와 과제수행행동에 미치는 영향. 미간행 석사학위논문, 단국대학교 대학원, 경기도.
- 김성준 (2019). 피지컬 컴퓨팅 기반 소프트웨어 교육이 초등학생의 컴퓨팅 사고력에 미치는 영향. 미간행 석사학위논문, 경인교육대학교 대학원, 인천.
- 김세용 (2002). 정렬 및 탐색 알고리즘의 효율적인 학습방안 연구. 미간행 석사학위논문, 인천대학교 교육대학원, 인천.
- 김소연, 민천식 (2019). 교수적합화를 통한 소프트웨어교육이 비장애학생의 장애 수용태도에 미치는 효과. **현장중심 초등교육연구**, 1, 257-276.
- 김순영 (2002). 중다예시 자기교시훈련이 발달장애유아의 과제수행 및 문제해결력의 일반화에 미치는 효과. **정서·행동장애연구**, 17(3), 79-106.
- 김시원 (2014). 보완대체 의사소통 사용을 위한 행동관찰 중재가 중도(重度)장애 학생의 자기결정행동에 미치는 영향. 미간행 박사학위논문, 조선대학교 대학원, 광주.
- 김시정, 조도는 (2018). 효과적인 코딩교육을 위한 학습 모델에 대한 연구. **한국융합학회논문지**, 9(2), 7-12.
- 김영기, 한희섭 (2011). 스프레드시트를 활용한 컴퓨터 프로그래밍 교육의 가능성에 관한 연구. **교육과학연구**, 42(2), 191-209.
- 김영표, 신현기 (2008). 장애학생의 수학적 문장제 문제해결에 관한 교수방법의 중재 효과: 메타 분석. **특수교육저널:이론과 실천**, 9(1), 413-437.
- 김완숙, 방명애 (2011). 교수적 루브릭 참조 자기평가가 통합된 장애학생의 수업참여행동과 비장애학생의 자기주도적 학습능력에 미치는 영향. **놀이치료연구**, 15(2), 111-126.
- 김인호, 유미현 (2019). 소프트웨어(SW) 창의교육이 초등학생의 창의적 문제해결력 및 SW 관련 진로지향도에 미치는 영향 및 성별에 따른 차이 분석. **실과교육연구**, 25(2), 151-177.
- 김재경 (2018). 소프트웨어 교육 연구의 동향 분석: 프로그래밍 언어 활용을 중심으로. **창의정보문화연구**, 4(1), 13-23.
- 김정일, 권미영 (2014). 마음챙김기반 중재가 지적장애 아동의 과제회피 및 과제

- 수행 행동에 미치는 효과: 사례연구. **특수교육재활과학연구**, 53(2), 169-191.
- 1.
- 김정일, 서영민, 이영준 (2012). 지적장애학생의 정보교육을 위한 언플러그드 학습 방법의 적용. **한국컴퓨터정보학회논문지**, 17(9), 189-196.
- 김철 (2019). 전통 민속놀이를 이용한 초등학교 언플러그드 컴퓨팅 교육-유희이를 중심으로. **정보교육학회논문지**, 23(6), 621-628.
- 김춘호 (2022). 초등 저학년 학생의 컴퓨팅 사고력 함양을 위한 비버챌린지 기반의 창의적 체험활동 프로그램 개발. 미간행 석사학위논문, 안동대학교 교육대학원, 안동.
- 김태용, 최하영 (2013). 장점강화활동이 지적장애 초등학생의 수업참여행동에 미치는 영향. **특수교육**, 12(1), 107-132.
- 김필현 (2021). 그림책을 활용한 언플러그드 교육이 유아의 컴퓨팅 사고력에 미치는 영향. 미간행 석사학위논문, 부산교육대학교 교육대학원, 부산.
- 김한성, 전수진 (2019). SW 교육을 위한 직무 및 정책에 대한 초등학교 교사들의 중요도와 수행도 인식 분석. **교육정보미디어연구**, 25(1), 151-170.
- 김현진 (2008). 경도정신지체학생의 문제해결에 있어서 귀인양식의 변화에 관한 연구. **특수교육연구**, 15(1), 73-87.
- 김현진, 강옥려 (2008). 통합학급 담당교사의 통합교육에 대한 인식 및 운영에 대한 분석 연구. **특수교육연구**, 15(2), 33-55.
- 김혜란, 최선영 (2019). 초등과학 수업에서 오조봇 코딩을 활용한 SW 융합교육프로그램의 개발과 적용. **초등과학교육**, 38(2), 234-243.
- 나원영, 이철현 (2016). 언플러그드 컴퓨팅을 활용한 STEAM 프로그램이 초등학생의 창의적 문제해결력에 미치는 영향. **실과교육연구**, 22(3), 79-95.
- 나청수, 주현, 이진주, 김동식 (2018). 컴퓨팅 사고력 향상을 위한 국내 SW 교육에 대한 메타분석. **교육공학연구**, 34(3), 775-815.
- 남수정 (2020). 엔트리를 활용한 초등 수학 코딩 학습에서 나타나는 문제해결 활동 및 수학적 태도의 특성. 미간행 석사학위논문, 서울교육대학교 대학원, 서울.
- 노지예, 이정민 (2017). 로봇 활용 SW 교육이 초등학생의 컴퓨팅사고력에 미치는 효과. **정보교육학회논문지**, 21(3), 285-296.
- 노지예, 이정민 (2018). 로봇 활용 SW 교육에서 컴퓨팅사고력 평가: 지필 시험,

- 비버챌린지, 자기보고식 설문을 중심으로. **교육공학연구**, 34(3), 849-876.
- 마대성 (2021). SW교육에 대한 초등교사와 학생의 인식, 태도 비교 분석 연구 - SW교육 선도학교와 일반학교를 대상으로-. **정보교육학회논문지**, 25(1), 185-193.
- 문병훈, 장천, 이영철 (2017). 개별차원의 긍정적 행동지원이 지적장애 초등학생의 문제행동에 미치는 영향. **지적장애연구**, 19(1), 105-126.
- 문영임, 이성규, 정희영, 최한주 (2021). 장애학생 SW교육에 대한 특수교사의 인식 및 현황에 대한 분석. **재활복지**, 25(4), 109-134.
- 문우중, 부용호, 김종훈 (2020). 수행시간 중심의 초등 EPL 교육이 초등학생 컴퓨팅 사고력에 미치는 효과. **정보교육학회논문지**, 24(3), 233-241.
- 민종선, 김우리 (2021) 맨드 토포그래피 사정을 통한 기능적 의사소통 훈련이 자폐성장애 학생의 자해해동과 맨드에 미치는 효과. **정서·행동장애연구**, 37(2), 29-46.
- 박선주 (2019). 알고리즘 교육을 통한 비버챌린지 결과 분석. **정보교육학회논문지**, 23(1), 65-72.
- 박신영, 손승현 (2010). 문제중심학습 (PBL)을 적용한 사회과 수업이 지적장애 학생들의 학업 성취도와 문제해결력에 미치는 영향. **통합교육연구**, 5(1), 1-20.
- 박정신, 조석봉 (2012). 프로그래밍입문 수업에서 스크래치 활용 효과분석. **디지털정책연구**, 10(9), 449-456.
- 박준석, 오정석 (2021). **어린이도 부모님도 알아야 할 코딩하기 전 코딩책**. 서울: 동아시아사이언스.
- 변관석 (2016). 연극놀이를 활용한 선행적 신체활동이 발달장애학생의 수업참여 행동에 미치는 효과. **특수교육연구**, 23(1), 232-250.
- 서민영, 정은희 (2021). 스토리텔링을 적용한 수학 수업이 지적장애 고등학생의 수학 학업성취도 및 수업 참여 행동에 미치는 영향. **한국청각·언어장애교육연구**, 12(2), 163-178.
- 서영호, 김종훈 (2018). CSCL 기반 디자인 사고를 적용한 SW교육이 초등학생의 창의성과 문제해결능력에 미치는 효과. **정보교육학회논문지**, 22(4), 427-438.
- 서영호, 염미령, 김종훈 (2016). 초등학교 SW교육에서 동료 프로그래밍 교육 방

- 법이 컴퓨팅 사고력과 창의성 신장에 미치는 효과 분석. **정보교육학회논문지**, 20(3), 219-234.
- 서현석, 정영식 (2018). 엔트리 프로그래밍 언어를 활용한 협력적 이야기 생성하기 활동의 교육적 효과. **정보교육학회논문지**, 22(6), 651-660.
- 서현주, 박은혜 (2018). 스위치를 활용한 AAC 중재가 지체·중복장애 고등학생의 수업참여행동에 미치는 영향. **지체·중복·건강장애연구**, 61(2), 1-27.
- 성영훈, 유승한 (2016). SCC 활동 전략기반 Kodu SW교육과정 모델 적용을 통한 어린이 코딩 인지 효과. **정보교육학회논문지**, 20(3), 283-292.
- 성정숙, 김현철 (2015). 국외 컴퓨터 교육과정의 변화 분석. **한국컴퓨터교육학회 논문지**, 18(1), 45-54.
- 송옥지, 박은경, 배종민 (2020). 마이크로비트를 활용한 소프트웨어 교육이 초등학생의 컴퓨팅사고력에 미치는 영향. **한국지식정보기술학회논문지**, 15(1), 37-46.
- 송의성, 임화경 (2021). 해외 SW·AI 교육 현황 분석을 통한 초등학교 정보 교과의 필요성. **정보교육학회논문지**, 25, (2), 301-308.
- 송준만, 강경숙, 김미선, 김은주, 김정효, 김현진, 이경순, 이금진, 이정은, 정귀순 (2012). **지적장애아교육**. 서울: 학지사.
- 송한민 (2021). 언플러그드 코딩활동에서 나타나는 만 5세 유아의 수학적 경험. 미간행 석사학위논문, 덕성여자대학교 대학원, 서울.
- 시지현 (2012). 적응적 교수프로그램과 멀티미디어를 활용한 과정 지향적 완성된 예제가 교수 효율성에 미치는 효과. **교육정보미디어연구**, 18(4), 417-440.
- 신성희, 한경근 (2019). 지적장애 고등학생의 알고리즘 학습과 사회성 기술을 위한 언플러그드 협동학습 사례 연구. **특수교육교과교육연구**, 12(4), 309-336.
- 심재권 (2019). 초중등 학습자의 알고리즘적 사고 수준 측정 연구. **창의정보문화연구**, 5(3), 237-243.
- 안성우, 서유경 (2019). 능동적 학습(Active Learning)이 예비특수교사의 청각장애교육 교과지식과 용어 습득에 미치는 영향. **한국청각·언어장애교육연구**, 10(2), 69-84.
- 안창호, 이기쁨, 문석재 (2018). 초보학습자를 위한 엔트리 블록/텍스트 코딩 기반의 프로그래밍 학습방법. **융복합지식학회논문지**, 6(1), 127-134.
- 양미경 (2007). 질문 창출 노력의 교육적 의의와 한계. **열린교육연구**, 15(2), 1-

- 20.
- 양지숙, 허유성 (2020). 소리-기호 간 연결 강화 중재가 읽기부진 학생의 읽기 유창성 및 과제회피 행동에 미치는 효과. **교과교육연구**, 41(2), 33-57.
- 오영범 (2017). 프로젝트 수업 사례를 통한 프로젝트 수업의 의미 탐색. **한국교육**, 44(1), 5-32.
- 오지훈, 장대원, 정일용 (2021). EPL 및 코딩 로봇 활용 SW교육이 컴퓨팅 사고. 과 문제해결력에 미치는 효과. **스마트미디어저널**, 10(3), 60-67.
- 유인환 (2016). 소프트웨어 공학 : 컴퓨팅사고력 신장을 위한 SW 개발 프로세스 탐구. **한국정보처리학회**, 5(2), 51-58.
- 유인환 (2018). 파이썬과 로봇을 활용한 초등학교 SW 교육의 설계. **정보교육학회 학술논문집**, 9(1), 149-155.
- 유지원 (2022). 앱 인벤터를 활용한 SW프로젝트 교육이 컴퓨팅 사고력 기반 문제 해결력과 SW학습 과제가치 인식에 미치는 효과: 인문사회계열 신입생을 대상으로. **컴퓨터교육학회 논문지**, 25(5), 27-36.
- 유환조 (2015). 긍정적 행동지원이 지적장애 중학생의 문제행동, 수업참여행동 및 인식에 미치는 영향. 미간행 박사학위논문, 우석대학교 대학원, 완주.
- 윤선희 (2018). EPL이 프로그래밍 교육의 프로그래밍, 컴퓨팅사고력 및 문제해결력에 미치는 영향에 관한 연구. *The Journal of the Convergence on Culture Technology*, 4(4), 287-294.
- 윤혁 (2016). Kagan의 구조중심 협동학습, 스마트러닝, 협동학습과 스마트러닝 통합중재가 경도장애 학생의 진로직업교육에 미치는 상대적 효과. 미간행 박사학위논문, 조선대학교 대학원, 광주.
- 이경순 (2007). 특수학급 교사들의 PBL(문제중심학습) 적용을 위한 교수적 지원이 장애 아동의 언어적 문제해결력 및 수업참여행동에 미치는 영향. 미간행 박사학위논문, 이화여자대학교 대학원, 서울.
- 이경희 (2020). 유아 소프트웨어 교육을 위한 교육과정 개발 연구. 미간행 박사학위논문, 제주대학교 대학원, 제주.
- 이명주, 박은혜, 김주혜 (2011). 그림상징 반응카드의 사용이 장애아동의 수업참여행동과 문제행동에 미치는 영향. **지적장애연구**, 13(3), 175-203.
- 이상철 (2012). 사회과 정착수업이 지적장애학생의 주의집중과 수업활동참여 및 과제수행에 미치는 영향. 미간행 석사학위논문, 한국교원대학교 교육대학원,

청주.

- 이선영, 강옥려 (2021). 계산기를 활용한 화폐교육이 지적장애 초등학생의 물건 값 계산하기와 수업참여도에 미치는 효과. **학습자중심교과교육연구**, 21(8), 241-255.
- 이소율, 이영준 (2017). 초등 교사의 정보 교수효능감 향상을 위한 EPL 교육 프로그램의 개발 및 적용. **한국컴퓨터교육학회논문지**, 20(5), 35-47.
- 이소현, 박은혜 (2011). **특수아동교육 3판**. 서울: 학지사.
- 이소현, 박은혜, 김영태(2000). **교육 및 임상현장 적용을 위한 단일대상연구**. 서울: 학지사.
- 이승연 (2019). 학습자의 인지부담 경감을 위한 효율적인 소프트웨어 교육모델 연구. 미간행 석사학위논문, 동국대학교 교육대학원, 서울.
- 이승원 (2019). 초등실과 SW교육에서 자기조절학습기반 언플러그드활동이 컴퓨팅 사고력에 미치는 효과. 미간행 박사학위논문, 충남대학교 대학원, 대전.
- 이승원, 이승연 (2017). 초등학교 5~6학년을 대상으로 한 공감각 융합인재교육(S TEAM) 프로그램. **예술인문사회융합멀티미디어논문지**, 7(2), 875-882.
- 이승원, 최유현 (2019). 초등실과 SW교육에서 자기조절학습기반 언플러그드활동이 컴퓨팅 사고력에 미치는 효과. **한국실과교육학회지**, 32(2), 105-121.
- 이신현 (2018). 컴퓨팅사고력 기반 소프트웨어교육이 초등학교 학생들의 학습 흥미와 진로인식에 미치는 영향. **초등교육학연구**, 25(1), 49-72.
- 이애화 (2018) 국내 소프트웨어교육 연구동향 분석. **교육정보미디어연구**, 24(2), 277-301.
- 이영준, 백성혜, 신재홍, 유현창, 정인기, 안상진, 최정원, 전성균 (2014). **초중 등 단계 Computational Thinking 도입을 위한 기초연구**. 서울: 한국과학창의재단.
- 이옥인, 백은희 (2005). 지적장애아동의 교수전략 동향에 대한 해외문헌분석 -1997년에서 2004년까지 인지 및 초인지 연구를 중심으로-. **정서·행동장애연구**, 21(3), 101-126.
- 이은경, 이영준 (2007). 로봇 프로그래밍 학습이 문제해결력에 미치는 영향. **한국컴퓨터교육학회논문지**, 10(6), 19-27.
- 이은경, 이영준 (2008). Scratch 활용 프로그래밍 교육이 중학생의 몰입수준과 프로그래밍 능력에 미치는 영향. **중등교육연구**, 56(2), 359-382.

- 이재은, 김준수 (2020). 북유럽 국가 사례를 중심으로 살펴본 유아 소프트웨어(SW) 교육 고찰. **유아교육연구**, 40(3), 229-251.
- 이재호, 김선향 (2021). 피지컬 컴퓨팅 기반 소프트웨어 교육이 초등학생의 컴퓨팅 사고력에 미치는 영향. **창의정보문화연구**, 7(4), 243-255.
- 이재호, 오상미 (2021). 초등저학년 학생을 대상으로 한 놀이학습 기반 언플러그드 교육프로그램 연구. **창의정보문화연구**, 7(2), 79-90.
- 이재호, 차근민 (2020). 블록코딩 SW 교육과 초등학생의 회복탄력성의 관계. **창의정보문화연구**, 6(1), 31-40.
- 이정민, 박현경, 최형신 (2018a). 로봇 활용 SW교육이 초등학생의 컴퓨팅사고력, 창의성, 학업흥미, 협업능력에 미치는 효과. **정보교육학회논문지**, 22(1), 9-21.
- 이정민, 채유정, 이명화 (2018b). 온라인 SW교육에서 초등학생의 컴퓨팅사고력 및 학습만족도에 대한 자기조절학습, 그릿, 부모지원의 예측력 규명. **정보교육학회논문지**, 22(6), 689-699.
- 이정민, 정제영 (2022). Birkland의 사건 중심 정책학습 모형을 활용한 AI·SW교육 정책변동 분석. **한국교육문제연구**, 40(3), 121-144.
- 이좌택, 이상봉 (2004). 컴퓨터 프로그래밍 학습에서 논리적 사고력 측정도구의 개발과 타당화 연구. **한국컴퓨터학회논문지**, 7(4), 15-25.
- 이준형, 김종석, 이형욱 (2018). 컴퓨팅 사고력 교육을 위한 아두이노 기반의 텐저블 유저 인터페이스 모듈 설계. **예술인문사회융합멀티미디어논문지**, 8(8), 451-459.
- 이철현 (2015). 초등 SW교육 방향 탐색 및 모델 개발. **한국실과교육학회지**, 28(4), 207-222.
- 이철현 (2017). 컴퓨팅 사고력 기반 실생활 문제해결학습이 초등학생의 컴퓨팅 사고력 향상에 미치는 효과. **실과교육연구**, 23(4), 91-107.
- 이철현 (2019). 초등예비교사의 EPL 프로그래밍 학습 어려움 분석. **한국실과교육학회지**, 32(2), 49-63.
- 이철현 (2021). EPL 프로그래밍의 어려움 및 오류 유형에 따른 맞춤형 모듈식 EPL 교육프로그램 개발 및 적용. **교육논총**, 41(3), 83-103.
- 이필애 (2007). 메타인지 학습 전략이 초등학교 수학학습장애 학생의 문장제 문제해결력과 자기효능감에 미치는 영향. 미간행 석사학위논문, 인제대학교 교

- 육대학원, 김해.
- 이현정 (2022). 특별한 관심 영역을 활용한 전략교수가 자폐 범주성 장애 학생의 문장제 수학 과제 수행률 및 정확도에 미치는 영향. 미간행 석사학위논문, 이화여자대학교 대학원, 서울.
- 이현주, 이원주, 정회경 (2020). A study on effective software education mode l by disability type for youth. **한국컴퓨터정보학회논문지**, 25(10), 261-268.
- 임해주 (2017). 기술·가정교과의 교수적 수정이 경도지적장애학생의 수업참여도 및 학습목표달성도에 미치는 영향. **행동분석·지원연구**, 4(2), 61-85.
- 임희석 (2017). **알기 쉬운 컴퓨팅 사고력**. 서울: 휴먼싸이언스.
- 장수정, 전현희, 강희숙, 주희정, 이서영, 신윤정, 김명윤, 유재혁 (2019). **컴퓨팅 사고력을 키우는 컴퓨터과학 알고리즘 스크래치 3.0**. 경기: (주)생능출판사.
- 장윤재, 김동형, 김한성, 이원규, 김현철 (2011). 정보보호 교육을 위한 언플러그드 활동의 개발 및 유용성 평가. **한국컴퓨터교육학회논문지**, 14(1), 55-67.
- 장재득, 주길홍 (2015). 알고리즘 학습을 통한 초등학생 문제해결력에 관한 연구. **창의정보문화연구**, 1(1), 39-47.
- 장정훈, 김종우 (2016). 초등학생용 문제해결력 증진을 위한 정렬 알고리즘 교육 자료 개발. **정보교육학회논문지**, 20(2), 151-160.
- 전석주 (2021). 절차적 프로그래밍 교육을 위한 텐저블 프로그래밍 도구 개발. **한국초등교육**, 32, 43-54.
- 전수진 (2017). SW 교육에서의 모듈 카드를 활용한 협동학습의 효과. **정보교육학회논문지**, 21(2), 191-198.
- 전혜진, 김승민, 이경현 (2016). 숨은그림찾기 그림책에 의한 주의집중력 효과 연구 -초등학교 저학년을 중심으로-. **커뮤니케이션디자인학회연구**, 57, 142-154.
- 정상목 (2020). 블록 코딩 온라인 저지시스템의 설계. **에듀테인먼트연구**, 2(1), 57-71.
- 정소영 (2019). 초등학교 컴퓨터 수업이 특수(지적장애)학생 교육의 학습동기 및 학업성취도에 미치는 효과. 미간행 석사학위논문, 인천대학교 교육대학원,

인천.

- 정웅열, 이영준 (2017). 정보 교육에서 비버 챌린지(Bebras Challenge)의 활용 가능성과 향후 과제. **한국컴퓨터교육학회논문지**, 20(5), 1-14.
- 정인재, 전석주 (2021). 엔트리 파이썬을 활용한 프로그래밍 학습이 초등학생의 논리적 사고력에 미치는 영향. **정보교육학회논문지**, 25(4), 603-610.
- 정지영 (2022). 개별차원의 긍정적 행동지원이 지적장애 초등학생의 수업 방해 행동과 수업 참여 행동에 미치는 영향. 미간행 석사학위논문, 단국대학교 대학원, 경기도.
- 정지은, 양문봉 (2021). 개별시도교수 중재 시 과제 선택하기 기법이 자폐스펙트럼장애 아동의 과제 회피행동에 미치는 효과. **행동분석·지원연구**, 8(2), 23-45.
- 정해원 (2019). 스마트폰 동영상상을 통한 시각적 피드백이 지적장애 플로어볼 참가자들의 슈팅 능력에 미치는 영향. 미간행 석사학위논문, 용인대학교 교육대학원, 경기도.
- 정희정 (2006). 경계선 지적 기능 아동의 특성 연구. 미간행 박사학위논문, 숙명여자대학교 대학원, 서울.
- 조미현 (2012). 신체활동이 지적장애학생의 주의집중력에 미치는 영향. 미간행 석사학위논문, 대구교육대학교 교육대학원, 대구.
- 조성환, 송정범, 김성식, 이경화 (2008). CPS에 기반한 스크래치 EPL이 문제해결력과 프로그래밍 태도에 미치는 효과. **한국정보교육학회**, 12(1), 77-88.
- 조은빛 (2022). 심리운동이 정서장애 아동의 주의집중에 미치는 영향. 미간행 석사학위논문, 나사렛대학교 대학원, 천안.
- 조은진 (2022). 피지컬 컴퓨팅을 활용한 소프트웨어 교육이 초등학생의 컴퓨팅 사고력에 미치는 영향. 미간행 석사학위논문, 경상국립대학교 교육대학원, 진주.
- 주현정, 이윤미 (2009). 어린이집 2세반 영아들의 자발적 질문에 관한 연구. **유아교육·보육행정연구**, 13(1), 31-64.
- 차재경, 김진호 (2014). 학급단위의 보편적 차원 긍정적 행동지원이 통합학급 학생들의 수업 참여행동과 방해행동에 미치는 영향. **지적장애연구**, 16(4), 85-109.
- 채성은, 허유성 (2021). 최근 10년간 국내외 특수교육 분야의 SW교육 연구 동향

- 분석. **학습장애연구**, 18(1), 127-153.
- 최수영, 송승하, 김영태 (2021). 초등학교 저학년 고기능 자폐 아동의 질문유형 및 자극유형에 따른 아이러니 이해 능력과 마음 이론과의 관계. *Communication Sciences and Disorders*, 26(1), 96-107.
- 최유현 (2019). 초등실과 SW교육에서 자기조절학습기반 언플러그드활동이 컴퓨팅 사고력에 미치는 효과. 미간행 박사학위논문, 충남대학교 대학원, 대전.
- 최유현, 이승원 (2019). 초등실과 SW교육에서 자기조절학습기반 언플러그드활동이 컴퓨팅 사고력에 미치는 효과. *한국실과교육학회지*, 32(2), 105-121.
- 한경근 (2017). 인공지능 테크놀로지 시대의 중도·중복장애학생 교육을 위한 제언. *지체·중복·건강장애연구*, 60(3), 47-65.
- 한경화, 김영옥 (2016). 경도장애 학생들의 수학적 문제해결을 위한 폴리아의 전략 효과 연구. *영남수학회*, 32(2), 253-289.
- 한국비버챌린지 (2019). **함께 즐기는 컴퓨터 사고와 정보과학 비버챌린지 -2019년도 기출문제집-**. 경기: 생능출판사.
- 한병래 (2013). 초등정보교육에서의 계산적사고 교육을 위한 언플러그드 컴퓨팅 방법에 대한 고찰. *한국정보교육학회논문지*, 17(2), 147-156.
- 한선관 (2010). 언플러그드 컴퓨팅과 EPL을 이용한 초등정보영재교육프로그램의 개발. *한국정보교육학회논문지*, 15(1), 31-38.
- 한선관, 김경신 (2007). 초등학생을 위한 컴퓨터과학의 언플러그드 학습 방법 연구. *정보교육학회논문지*, 11(4), 497-504.
- 한선관, 김수환 (2010). EPL을 활용한 수학문제해결 통합교육프로그램의 학년 수준 비교. *한국정보교육학회논문지*, 14(3), 311-318.
- 한옥영, 김재현 (2011). 효과적인 알고리즘 교육을 위한 교수-학습 모형 개발. *컴퓨터교육학회논문지*, 14(2), 13-22.
- 한지원 (2019). 프로그래밍언어를 활용한 유아 소프트웨어교육 실행과정에서 겪은 유아의 경험. 미간행 석사학위논문, 한국교원대학교 대학원, 청주.
- 한지원, 최연철 (2018). 프로그래밍언어를 활용한 유아 소프트웨어교육 실행 과정에서 겪은 교사의 경험. *유아교육연구*, 38(6), 91-115.
- 허영 (2019). 초등학교 언플러그드 코딩교육을 위한 프로그램 개발. *기초조형학연구*, 20(1), 586-597.
- 허희옥, 임규연, 서정희, 김영애(2011). 21세기 학습자 및 교수자 역량 모델링.

대구: 한국교육학술정보원.

- 홍다희 (2016). EPL을 활용한 SW교육이 경도 지적장애 아동의 인지과정에 미치는 영향에 관한 사례연구. 미간행 석사학위논문, 단국대학교 대학원, 경기도.
- 홍성희 (2003). 비디오를 이용한 자기모델링 중재가 자폐아동의 질문하기 기능 습득에 미치는 효과. 미간행 석사학위논문, 이화여자대학교 대학원, 서울.
- 홍순재 (2019). 지적장애학생의 절차적 사고 함양을 위한 모바일 학습 프로그램 개발 및 적용. 미간행 석사학위논문, 서울교육대학교 교육전문대학원, 서울.
- 황규용 (2017). 지적장애학생에게 피지컬컴퓨팅 도구 적용 시 나타난 교육적 효과의 질적 연구 분석. 미간행 석사학위논문, 경인교육대학교 대학원, 인천.
- 황성신 (2019). 소프트웨어 교육 활동 과정에서 나타난 유아의 컴퓨팅 사고 경험. 미간행 석사학위논문, 한국교원대학교 대학원, 청주.
- 황지은, 황성온 (2017). 초등 소프트웨어 교육 연구 동향 분석: 국내 학술지 중심으로. **정보교육학회논문지**, 21(5), 509-525.
- Alberto, P., Troutman, A. C. (2010). **교사를 위한 응용행동분석**. 서울: 학지사.
- APA (2018). **DSM-5 간편 정신질환진단통계편람**. 서울: 학지사.
- Ayer, T. (2018). *Relationships Between Learner Characteristics and Computational Thinking for Middle-School Students in Two Different Contexts* (Doctoral dissertation, Doctoral dissertation).
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- Bell, T., & Vahrenhold, J. (2018). CS unplugged—how is it used, and does it work?. *Adventures between lower bounds and higher altitudes*(pp. 497-521). Springer, Cham.
- Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20-29.
- Bippus, A. M., & Young, S. L. (2000). What behaviors reflect involvement in a course?: Students' perceptions and differences between high and low c

- ommunication apprehensives. *Communication Research Reports*, 17(3), 310–319.
- Bouck, E. C., & Yadav, A. (2022). Providing access and opportunity for computational thinking and computer science to support mathematics for students with disabilities. *Journal of Special Education Technology*, 37(1), 151–160.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65–72).
- da Silva, F. F., Aylon, L. B. R., & Flôr, D. E. (2020, October). Teaching computational thinking to a student with attention deficit through programming. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.
- Dietz, G., Landay, J. A., & Gweon, H. (2019). Building blocks of computational thinking: Young children's developing capacities for problem decomposition. In *CogSci* (pp. 1647–1653).
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28.
- Fassinger, P. A. (1996). Professors' and students' perceptions of why students participate in class. *Teaching sociology*, 25–33.
- Franklin, D., Salac, J., Thomas, C., Sekou, Z., & Krause, S. (2020). Eliciting student Scratch script understandings via Scratch Charades. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 780–786.
- Furrer, C., & Skinner, E. (2003). Sense of relatedness as a factor in children's academic engagement and performance. *Journal of educational psychology*, 95(1), 148.

- Futschek, G. (2006, November). Algorithmic thinking: the key for understanding computer science. In *International conference on informatics in secondary schools—evolution and perspectives* (pp. 159–168). Springer, Berlin, Heidelberg.
- Gomes, A., & Mendes, A. J. (2007, September). Learning to program—difficulties and solutions. In *International Conference on Engineering Education -ICEE* (Vol. 7).
- González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., & Infante-Moro, A. (2019, June). Computational thinking and down syndrome: An exploratory study using the KIBO robot. In *Informatics* (Vol. 6, No. 2, p. 25). MDP I.
- Grey, I. M., & Hastings, R. P. (2005). Evidence-based practices in intellectual disability and behaviour disorders. *Current opinion in psychiatry*, *18*(5), 469–475.
- Horn, M. S., & Jacob, R. J. (2007, February). Designing tangible programming languages for classroom use. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (pp. 159–162).
- Hufad, A., Faturrohman, M., & Rusdiyani, I. (2021). Unplugged Coding Activities for Early Childhood Problem-Solving Skills. *Jurnal Pendidikan Usia Dini*, *15*(1), 121–140.
- Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. (2015a). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *TEACHING Exceptional Children*, *48*(1), 45–53.
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015b). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263–279.
- Israel, M., Ray, M. J., Maa, W. C., Jeong, G. K., eun Lee, C., Lash, T., & Do, V. (2018). School-embedded and district-wide instructional coaching in K-8 computer science: Implications for including students with disab

- ilities. *Journal of Technology and Teacher Education*, 26(3), 471-501.
- Lockwood, J., & Mooney, A. (2017). Computational thinking in education: Where does it fit? A systematic literary review. *arXiv preprint arXiv:1703.07659*.
- Mize, M., & Park, Y. (2021). iPad-assisted reading fluency instruction for fourth graders with reading difficulties: A single case experimental design. *Journal of Computer Assisted Learning*, 37(2), 500-509.
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, (46), 1-23.
- Moschella, M. (2019). Observable computational thinking skills in primary school children: How and when teachers can discern abstraction, decomposition and use of algorithms. *challenge*, 16, 17.
- Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, A., & Bustos-Valenzuela, P. (2018). Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. *IEEE Access*, 6, 63880-63889.
- Mustapha, S. M., Abd Rahman, N. S. N., & Yunus, M. M. (2010). Perceptions towards classroom participation: A case study of Malaysian undergraduate students. *Procedia-Social and Behavioral Sciences*, 7, 113-121.
- Pakarinen, E., Kiuru, N., Lerkkanen, M. K., Poikkeus, A. M., Ahonen, T., & Nurmi, J. E. (2011). Instructional support predicts children's task avoidance in kindergarten. *Early Childhood Research Quarterly*, 26(3), 376-386.
- Polya, G. (1962). *Mathematical discovery*, 1962. John Wiley & Sons.
- Resnick, M. (2007, June). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*(pp. 1-6).
- Rhode, G., Morgan, D. P., & Young, K. R. (1983). GENERALIZATION AND MAINTENANCE OF TREATMENT GAINS OF BEHAVIORALLY HANDICAPPED STUDENTS FROM RESOURCE ROOMS TO REGULAR CLASSROOMS USING SELF-EVALUATION PROCEDURES. *Journal*

- J of applied behavior analysis*, 16(2), 171-188.
- Robert, L. S., Ruth, L., & Marc, J. T. (2022). **AAIDD 12판 지적장애: 정의, 진단, 분류, 및 지원체계**. 서울: 교육학지사.
- Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017, March). Assessing computational thinking in CS unplugged activities. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 501-506).
- Román-González, M., Moreno-León, J., & Robles, G. (2017, July). Complementary tools for computational thinking assessment. In *Proceedings of International Conference on Computational Thinking Education (CTE 2017)*.
- Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2016, November). Does computational thinking correlate with personality? The non-cognitive side of computational thinking. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 51-58).
- Ronfard, S., Zambrana, I. M., Hermansen, T. K., & Kelemen, D. (2018). Question-asking in childhood: A review of the literature and a framework for understanding its development. *Developmental Review*, 49, 101-120.
- Settles, B. (2009). Active learning literature survey.
- Shin, S., & Park, P. (2014). A study on the effect affecting problem solving ability of primary students through the scratch programming. *Advanced Science and Technology Letters*, 59(1), 117-120.
- Taylor, M. S. (2018). Computer programming with Pre-K through first-grade students with intellectual disabilities. *The Journal of Special Education*, 52(2), 78-88.
- Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer programming with early elementary students with Down syndrome. *Journal of Special Education Technology*, 32(3), 149-159.
- Tu, J. J., & Johnson, J. R. (1990). Can computer programming improve problem-solving ability?. *ACM SIGCSE Bulletin*, 22(2), 30-33.
- Voskoglou, M. G., & Buckley, S. (2012). Problem solving and computational t

- hinking in a learning environment. *arXiv preprint arXiv:1212.0750*.
- Vourletsis, I., Politis, P., & Karasavvidis, I. (2021). The Effect of a Computational Thinking Instructional Intervention on Students' Debugging Proficiency Level and Strategy Use. In *Research on E-Learning and ICT in Education* (pp. 15–34). Springer, Cham.
- White, G. L., & Sivitanides, M. P. (2002). A theory of the relationships between cognitive requirements of computer programming languages and programmers' cognitive characteristics. *Journal of information systems education, 13*(1), 59.
- Wille, S., Century, J., & Pike, M. (2017). Exploratory research to expand opportunities in computer science for students with learning differences. *Computing in Science & Engineering, 19*(3), 40–50..
- 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 A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717–3725.
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all I: Pedagogical approaches to embedding 21st century problem solving in K–12 classrooms. *TechTrends, 60*(6), 565–568.
- Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019, March). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. In *Society for information technology & teacher education international conference* (pp. 347–350). Association for the Advancement of Computing in Education (AACE).

부록

[부록1] 프로그램 참가 동의서

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시

[부록3] 문제해결력 관찰기록표

[부록4] 수업 참여행동 관찰기록표

[부록5] 중재 충실도

[부록6] 사회적 타당도

[부록1] 프로그램 참가 동의서

연구참여자 동의서	
<p>기간: 2022년 4월 ~ 2022년 7월</p> <p>의뢰인: 조선대학교 일반대학원 특수교육학과 채성은</p> <p>Tel: 010-0000-0000 E-mail: 0000000000@gmail.com</p>	
<p>안녕하십니까?</p> <p>조선대학교 일반대학원 특수교육학과 박사과정생 채성은입니다.</p> <p>이 연구는 언플러그드 교육과 EPL 교육이 장애학생의 컴퓨팅 사고력, 문제 해결력, 수업 참여행동에 미치는 영향을 알아보기 위한 것입니다. 이 연구는 장애학생의 SW교육에 중요한 정보를 제공할 것으로 기대하고 있습니다.</p>	
<ol style="list-style-type: none"> 1. 연구 기간: 2022년 4월 ~ 7월 2. 연구 내용 <ul style="list-style-type: none"> - 연구 회기: 주 2~3회 씩 총 24회기 수업 (1회 40분) - 관찰 내용: 언플러그드 교육과 EPL 교육 수업 모습 - 관찰 방법: 고정된 위치에 비디오 설치 후 녹화 및 녹음 - 수업 장소: 00초등학교 특수학급 - 수업 시간: 방과후 3. 수집 자료 보관 및 사용방법 <ul style="list-style-type: none"> - 수집된 자료는 연구자의 연구 이외의 다른 용도로 사용되지 않습니다. - 학생에 대한 정보는 모두 가명 처리할 것입니다. - 모든 자료는 연구 종료 후 폐기할 것입니다. 4. 언제라도 연구의 참여를 철회할 권리가 있습니다. 철회할 경우 철회 이전에 이루어진 관찰내용은 연구에서 제외되고 자료는 폐기할 것입니다. 	
연구 참여 동의서	
<p>2022년 4월 ~ 7월까지 실시되는 언플러그드 교육과 EPL 교육에 귀하의 자녀가 참여하는 것을 동의합니다.</p>	
<p>학년 반 학생이름 :</p>	
<p>2022년 월 일 보호자 :</p>	<p>(인)</p>

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시

주제	생활 속 소프트웨어	회기 (시간)	기초선 1회기 (40분)
학습목표	자신이 경험한 소프트웨어를 말할 수 있다.		
단계 (시간)	교수-학습활동		자료
활동 과정 및 내용	도입 (5)	<ul style="list-style-type: none"> ● ‘소프트웨어’ 라는 단어를 들어 본 적이 있는지 발표하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 5px; text-align: center; margin: 5px 0;">자신이 경험한 소프트웨어를 말해보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 	
	전개 (30)	<p><활동1> 소프트웨어의 뜻 알기</p> <ul style="list-style-type: none"> - ‘일상생활 속의 소프트웨어’ 영상을 보며 소프트웨어가 무엇인지 알아보기 - 하드웨어와 소프트웨어의 차이점 알아보기 <p><활동2> 소프트웨어의 하는 일 알기</p> <ul style="list-style-type: none"> - 네비게이션, 엘리베이터, 공기청정기, 전기밥솥 중 자신이 경험한 기기 선택하기 - 자신이 경험한 기기를 하나씩 선택하여 버튼을 눌렀을 때 어떻게 작동했는지 발표하기 <p><활동3> 편리한 소프트웨어</p> <ul style="list-style-type: none"> - 핸드폰, 엘리베이터, 세탁기, 에어컨이 작동하지 않을 때 생기는 어려움 발표하기 	<p><input checked="" type="checkbox"/> ebs ‘일상생활 속의 소프트웨어’ 동영상</p> <p><input checked="" type="checkbox"/> 생활 속 기기 사진 ppt</p>
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 소프트웨어가 하는 일을 읽고 알맞은 기기에 연결하기 	
유의점	<ul style="list-style-type: none"> • 소프트웨어가 사람의 별도 조작 없이 기기를 자동으로 움직일 수 있도록 명령하는 프로그래밍을 강조한다. • 제시된 기기 및 소프트웨어 외에도 다양하고 자유롭게 발표할 수 있도록 한다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제	컴퓨터의 입력장치와 출력 장치	회기 (시간)	기초선 2회기 (40분)
학습목표	컴퓨터의 입력장치와 출력장치를 구분할 수 있다.		
단계 (시간)	교수-학습활동		자료
활동 과정 및 내용	도입 (5)	<ul style="list-style-type: none"> ● 음료수 자판기 사용경험 이야기 나누기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">컴퓨터의 입력장치와 출력장치를 구분해보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 	<ul style="list-style-type: none"> ☐ 자 판 기 사진 PPT
	전개 (30)	<p><활동1> 생활 속 입력과 출력</p> <ul style="list-style-type: none"> - 입력과 출력의 용어 정의하기 - 생활 속에서 리모컨을 사용한 경험 이야기 나누고, 입력과 출력 구분하기 <p><활동2> 컴퓨터의 입력장치와 출력장치</p> <ul style="list-style-type: none"> - 컴퓨터의 주변기기 이름 알아보기 - 컴퓨터 주변기기의 작동법 발표하고, 입력장치와 출력장치로 구분해보기 <p><활동3> 입력하고 출력하기</p> <ul style="list-style-type: none"> - 음악 틀기, 프린트 하기, 키보드 입력하기 등의 입력과 출력활동하기 	<ul style="list-style-type: none"> ☐ 리모컨을 사용하는 기기 사진 ppt ☐ 컴퓨터 주변기기 사진 ppt ☐ 노트북, 모니터, 키보드, 프린트, 오디오
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 컴퓨터 입력과 출력장치 연결하기 	
유의점	<ul style="list-style-type: none"> • 입력 장치와 출력 장치가 어떻게 사용되는 지에 초점을 맞춘다. • 입력 장치와 출력 장치의 용어를 정의하고, 학생의 발표 내용에서 입력과 출력이 무엇이었는지 지도한다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제	태블릿 PC 사용법 알기	회기 (시간)	기초선 3회기 (40분)
학습목표	태블릿 PC로 필요한 정보를 검색할 수 있다.		
단계 (시간)	교수-학습활동		자료
활동 과정 및 내용	도입 (5)	<ul style="list-style-type: none"> ● 여러 가지 퀴즈를 보고 답을 어떻게 찾을지 고민하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">태블릿 PC로 필요한 정보를 검색해보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 	<ul style="list-style-type: none"> ☐ 퀴즈 PPT
	전개 (30)	<p><활동1> 스마트폰, 태블릿 PC 경험 이야기 나누기 - 스마트폰이나 태블릿 PC를 사용했던 경험을 알아보고, 어떤 활동을 했는지 발표하기</p> <p><활동2> 퀴즈 풀기 - 인터넷 브라우저 소개하고 검색방법 알아보기 - 활동지의 문제를 보고 검색하여 해답찾기</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <ol style="list-style-type: none"> 1. 나라의 국기를 검색하고 색칠하기 (독일, 프랑스) 2. 우리 학교 검색하여 학교 사진 찾기 3. 우리 학교 홈페이지 주소 검색하기 4. 우리 학교 홈페이지에서 교무실 전화번호 찾기 </div> <p>- 서로의 답 맞춰보기</p>	<ul style="list-style-type: none"> ☐ 스마트폰, 태블릿 PC, 사진 PPT ☐ 태블릿 PC, 퀴즈 활동지
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 퀴즈의 답을 어떻게 찾았는지 발표하기 	
유의점	<ul style="list-style-type: none"> • 엔트리 플랫폼에 접속할 때 사용 할 태블릿 PC에 대한 기초 지식을 습득하게 한다. • 퀴즈의 답 보다는 태블릿 PC 사용법에 초점을 맞춘다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제		규칙 찾기	회기 (시간)	언플러그드 1회기 (40분)
학습목표		물체의 배열에서 규칙을 찾을 수 있다.		
컴퓨팅 사고력		반복 나열, 패턴 이해		
단계 (시간)		교수-학습활동		자료 및 유의점
활동 과정 및 내용	도입 (5)	<ul style="list-style-type: none"> ● 생활 속 사진을 보며 어떤 규칙이 있는지 생각하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">규칙을 정해 가랜드를 만들어 보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		<input checked="" type="checkbox"/> 모양, 무늬, 색이 반복되는 생활사진 PPT
	전개 (30)	<p><활동1> 반복되는 규칙찾기</p> <ul style="list-style-type: none"> - 규칙 찾는 방법을 알아보기 - 제시한 규칙을 보고 다음에 올 도형 맞추기 <p><활동2> 사탕 가랜드 만들기</p> <ul style="list-style-type: none"> - 사탕 가랜드 만드는 방법 알아보기 - 3가지 사탕을 사용해 규칙을 정하고 가랜드 만들기 - 자신이 만든 사탕 가랜드의 규칙 발표하기 <p><활동3> 롤러코스터 타기 미션 해결하기</p> <ul style="list-style-type: none"> - 롤러코스터 타기 미션1 해결하기 - 자신이 세운 알고리즘 발표하기 		<input checked="" type="checkbox"/> 삼각형, 사각형, 원 자석교구 <input checked="" type="checkbox"/> 세가지 맛 사탕, 끈 <input checked="" type="checkbox"/> 알고리즘 판, 명령어 카드, 문제 ppt
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 숫자와 그림이 반복되는 문제 제시하고 규칙찾기 		<input checked="" type="checkbox"/> 퀴즈 ppt
유의점		<ul style="list-style-type: none"> • 규칙을 설명할 때 반복되는 부분을 덩어리짓고, ‘반복’, ‘되풀이된다’ 라는 용어를 사용할 수 있도록 유도한다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제		절차적 사고 이해	회기 (시간)	언플러그드 6회기 (40분)
학습목표		화분 설명서를 만들고 설명서 순서대로 화분을 만들 수 있다.		
컴퓨팅 사고력		규칙, 절차적 사고		
단계 (시간)		교수-학습활동		자료 및 유의점
활동 과정 및 내용	도입 (5)	<ul style="list-style-type: none"> ● 아빠로봇이 왜 샌드위치를 만들지 못했는지 알아보기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 2px; display: inline-block;">설명서를 보고 화분을 순서대로 만들어보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		☑ 동영상
	전개 (30)	<p><활동1> 화분 설명서 만들기</p> <ul style="list-style-type: none"> - 교사의 화분 만들기 설명서 제시하고 설명서 만드는 방법에 대해 알기 - 자신이 만들고 싶은 화분을 고르고 화분 설명서 만들기 - 자신의 화분 만들기 설명서 발표하기 <p><활동2> 순서대로 만들기</p> <ul style="list-style-type: none"> - 준비뽑기로 친구의 설명서 뽑기 - 친구의 설명서를 보고 순서대로 화분 만들기 - 자신이 만든 화분 발표하기 - 친구가 자신의 설명서대로 화분을 만들었는지 평가하기 <p><활동3> 누리호 발사 미션 해결하기</p> <ul style="list-style-type: none"> - 누리호 발사 미션6 해결하기 - 자신이 세운 알고리즘 발표하기 		☑ 선생님의 화분 만들기 설명서, 화분 만들기 설명서 활동지, 화분만들기 키트 ☑ 완성 화분 ☑ 알고리즘 판, 명령어 카드, 문제 ppt
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 설명서 만들기과 실제로 만들 때의 주의점 발표하기 		☑ 잘못된 순서로 작성된 순서도 PPT
유의점		<ul style="list-style-type: none"> • 자신이 구상한 화분의 모습과 친구가 만든 화분의 모습이 같지 않을 때 설명서를 보며 이유를 찾게 하고, 설명서를 자세하게 써야 함을 알 수 있게 하여 명령어의 중요성을 알게 한다. • 학습 정리에서 설명서를 순서대로 쓰지 않거나, 만들 때 순서를 따르지 않으면 원하는 작품이 나오지 않음을 알 수 있게 한다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제		코딩으로 만드는 자기소개	회기 (시간)	EPL 3회기 (40분)
학습목표		코딩으로 자기소개 화면을 만들 수 있다.		
컴퓨팅 사고력		절차적 사고, 알고리즘(순차)		
단계 (시간)		교수-학습활동		자료
활동 과정 및 내용	도입 (7)	<ul style="list-style-type: none"> ● 전시학습 상기하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 5px; margin: 10px 0; text-align: center;">엔트리로 자기 소개를 해보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		<input checked="" type="checkbox"/> 자기 소개 만들기 영상
	전개 (30)	<p><활동1> 자기 소개 하는 법 알기</p> <ul style="list-style-type: none"> - 스터디 학습하기로 들어가 목표 작품을 눌러 영상 확인하기 - ‘자기 소개 만들기’ 영상에 나온 오브젝트 찾아 열기 - 영상에 나온 장면을 기억하여 조립소에서 블록 찾아 조립하기 - 블록에 알맞은 대사 입력하기 <p><활동2> 나의 특징이 들어간 자기 소개 하기</p> <ul style="list-style-type: none"> - 내가 좋아하는 것, 나를 상징하는 이미지를 생각해 여러 가지 오브젝트 선택하고, 블록 조립하기 - 소개 글 3가지 이상 쓰기 - 엔트리로 자기 소개 발표하기 <p><활동3> 이상한 숲 속의 엔트리봇 미션 해결하기</p> <ul style="list-style-type: none"> - 이상한 숲 속의 엔트리봇 미션3 해결하기 - 자신이 세운 알고리즘 발표하기 		<input checked="" type="checkbox"/> 태블릿 PC <input checked="" type="checkbox"/> 태블릿 PC <input checked="" type="checkbox"/> 태블릿 PC
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 학습내용 정리하고 알게 된 점, 어려웠던 점 이야기 나누기 		
	유의점	<ul style="list-style-type: none"> • 자신을 나타내는 오브젝트를 선택할 수 있게 하고 자신을 나타내는 알맞은 대사를 쓸 수 있도록 한다. • 어떤 블록을 조립해야 할지 스스로 탐구할 수 있도록 한다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제		바닷 속 친구들 모습 만들기	회기 (시간)	EPL 7회기 (40분)
학습목표		반복 블록을 사용해 명령을 내릴 수 있다.		
컴퓨팅 사고력		반복 나열, 패턴 이해, 알고리즘(순차, 반복)		
단계 (시간)		교수-학습활동		자료
활동 과정 및 내용	도입 (7)	<ul style="list-style-type: none"> ● 전시학습 상기하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;">반복 블록을 사용해 명령을 내려보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		☞ 바닷 속 친구들 완성 화면 영상
	전개 (30)	<p><활동1> 바닷 속 친구들 모습 만들기</p> <ul style="list-style-type: none"> - 스터디 학습하기로 들어가 목표 작품을 눌러 영상 확인하기 - ‘바닷 속 친구들 모습 만들기’ 영상에 나온 오브젝트 찾아 열기 - 반복되는 행동 찾기 - 물고기가 화면 끝에 가면 다시 되돌아 오는 방법 찾기 - 반복 블록을 사용하여 블록 조립하기 - 완성작 발표하기 <p><활동2> 이상한 숲 속의 엔트리봇 미션 해결하기</p> <ul style="list-style-type: none"> - 이상한 숲 속의 엔트리봇 미션7 해결하기 - 자신이 세운 알고리즘 발표하기 		☞ 태블릿 PC, 바닷 속 친구들 완성 화면 영상 ☞ 태블릿 PC
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 학습내용 정리하고 알게 된 점, 어려웠던 점 이야기 나누기 		
유의점		<ul style="list-style-type: none"> • 같은 동작을 여러 번 움직이게 하고 싶을 때는 ‘반복’ 블록을 사용하게 한다. • 어떤 블록을 조립해야 할지 스스로 탐구할 수 있도록 한다. 		

[부록2] 기초선, 언플러그드 교육, EPL 교육 수업지도안 예시(계속)

주제		신비한 블록과 놀기	회기 (시간)	효과 2회기 (40분)
학습목표		도장 찍기 블록, 소리 블록을 사용해 명령을 내릴 수 있다.		
컴퓨팅 사고력		반복 나열, 패턴 이해, 알고리즘(순차, 반복)		
단계 (시간)		교수-학습활동		자료
활동 과정 및 내용	도입 (7)	<ul style="list-style-type: none"> ● 전시학습 상기하기 ● 학습문제 제시하기 <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">도장 찍기, 소리 블록을 사용해 명령을 내려보자</div> <ul style="list-style-type: none"> ● 학습순서 확인하기 		
	전개 (30)	<p><활동1> 바닷 속 친구들 모습 만들기</p> <ul style="list-style-type: none"> - 스테디 학습하기로 들어가 목표 작품을 눌러 영상 확인하기 - ‘사막에 선인장 심기’ 영상에 나온 오브젝트 찾아 열기 - 어떠한 행동이 반복되는지 발표하기 - 도장 찍기 블록, 반복 블록을 사용하여 블록 조립하기 - 완성작 발표하기 <p><활동2> 여러 가지 소리와 놀기</p> <ul style="list-style-type: none"> - 스테디 학습하기로 들어가 목표 작품을 눌러 영상 확인하기 - ‘여러 가지 소리와 놀기’ 영상에 나온 오브젝트 3가지 찾아 열기 - 오브젝트에 어울리는 소리 찾기 - 오브젝트와 소리를 연결하고 반복 블록 사용해 블록 조립하기 - 완성작 발표하기 		<p>☞ 태블릿 P C, 오아시스 완성 화면 영상</p> <p>☞ 태블릿 P C, 여러 가지 악기 완성 화면 영상</p>
	정리 (5)	<ul style="list-style-type: none"> ● 학습정리하기 - 학습내용 정리하고 알게 된 점, 어려웠던 점 이야기 나누기 		
유의점		<ul style="list-style-type: none"> • 서로의 작품을 감상하며 어떤 소리인지 맞춰보기 • 어떤 행동이 반복되는지 스스로 탐구할 수 있도록 한다. 		

[부록3] 문제해결력 관찰기록표

문제해결력 관찰기록표

학생명		회기/날짜				
도전횟수	시작시간	끝나는 시간	걸린시간 <small>(끝나는시간-시작시간)</small>	정답/ 오답	반복문 사용 여부	오류 점검 여부
		합계				
		평균				

[부록4] 수업 참여행동 관찰기록표-주의집중, 과제회피

주의집중, 과제회피 관찰기록표

학생명			회기/날짜					
측정방법			15초 관찰, 5초간 기록			관찰시간(시작~종료)		
기록방법			시간 간격 내에 한 번 이상 발생 시 + 표시					
구간	주의 집중	과제 회피	구간	주의 집중	과제 회피	구간	주의 집중	과제 회피
1분	0~15		11분	0~15		21분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
2분	0~15		12분	0~15		22분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
3분	0~15		13분	0~15		23분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
4분	0~15		14분	0~15		24분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
5분	0~15		15분	0~15		25분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
6분	0~15		16분	0~15		26분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
7분	0~15		17분	0~15		27분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
8분	0~15		18분	0~15		28분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
9분	0~15		19분	0~15		29분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	
10분	0~15		20분	0~15		30분	0~15	
	20~35			20~35			20~35	
	40~55			40~55			40~55	

- 주의집중행동 발생률: ()/90 × 100 = ()%
- 과제회피행동 발생률: ()/90 × 100 = ()%

[부록4] 수업 참여행동 관찰기록표- 자발적 질문

자발적 질문 관찰기록표

회기	날짜	수어유형	학생A		학생B		학생C	
			횟수(正)	계	횟수(正)	계	횟수(正)	계
기초선1								
기초선2								
기초선3								
중재1								
중재2								
중재3								
중재4								
중재5								
중재6								
중재7								
중재8								
중재9								
중재10								
중재11								
중재12								
중재13								
중재14								
중재15								
중재16								
효과1								
효과2								
효과3								

[부록5] 중재 충실도

회기		교육방법		
항목	문항	실행여부		
		예	아니오	
중재준비	1	행동 관찰을 위한 동영상 촬영 장비를 준비하였는가?		
	2	수업 전에 태블릿 PC 혹은 학습준비물을 미리 준비하였는가?		
중재절차	3	예정된 회기 순서에 맞게 중재 프로그램을 수행하였는가?		
	4	학생들이 수행할 수 있도록 시범을 보였는가?		
	5	각 차시 지도안에 따라 활동을 모두 수행하였는가?		
	6	활동을 마친 후 평가시간을 가졌는가?		
중재성실도	7	학생들의 결과물에 대한 적절한 반응을 보여주었는가?		
	8	학생들이 반성적 사고를 할 수 있도록 최소한의 촉진을 제공하였는가?		
	9	학생들이 활동 할 시간을 충분히 주었는가?		
	10	학생들이 활동에 80%이상 참여하였는가?		

[부록6] 사회적 타당도

항목	문항	5	4	3	2	1
중재목표의 중요성	장애학생들에게 소프트웨어 교육이 필요하다고 생각하는가?					
중재의 수용성	소프트웨어교육을 어렵지 않게 할 수 있을 것 같은가?					
	다른 교사에게도 소프트웨어교육을 추천해주고 싶은가?					
	소프트웨어교육은 장애학생에게 적절하였다고 생각하는가?					
중재 결과의 중요성	소프트웨어교육은 장애학생들에게 유익한가?					

척도

5: 매우 그렇다

4: 그렇다

3: 보통이다

2: 그렇지 않다

1: 전혀 그렇지 않다