



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](#)

교육학박사 학위논문

창의적 문제해결을 위한  
교과-소프트웨어 융합  
수업설계 모형 개발 연구

2021년 8월

서울대학교 대학원  
교육학과 교육공학전공  
지 현 경

창의적 문제해결을 위한  
교과-소프트웨어 융합  
수업설계 모형 개발 연구

지도교수 임 철 일

이 논문을 교육학박사 학위논문으로 제출함  
2021년 8 월

서울대학교 대학원  
교육학과 교육공학전공  
지 현 경

지현경의 박사 학위논문을 인준함  
2021년 8 월

위 원 장 \_\_\_\_\_

부위원장 \_\_\_\_\_

위 원 \_\_\_\_\_

위 원 \_\_\_\_\_

위 원 \_\_\_\_\_

## 국문초록

4차 산업혁명과 과학기술의 발달에 따라 기존과 다른 새로운 역량이 요구되고 있다. 미래에는 인공지능의 발달로 인한 일자리가 감소되는 반면 컴퓨터 알고리즘이 발달하면서 새로운 일자리를 창출할 것으로 전망되어 신산업을 이해하고 활용할 수 있는 소프트웨어와 관련한 지식 및 기술과 더불어 창의적 문제해결력, 융합적 사고, 비판적 사고와 같은 새로운 역량에 대한 요구가 증가할 것이며, 이를 갖춘 인재양성 교육이 필요하다. 미래 사회의 인재양성을 위한 노력의 일환으로 2015년부터 소프트웨어 교육 과정이 초·중·고등학교에 정규 과정으로 적용되었다. 일상생활의 복잡한 문제를 창의적으로 해결하는 과정을 통해 미래 사회의 창의적 융복합 인재 육성이 소프트웨어 교육 과정의 목표이다.

그러나 소프트웨어 교육은 미래 사회를 대비한 새로운 장점과 가능성을 가지고 있음에도 불구하고, 다른 교과와의 연계성이 부족하고 유기적인 관련성이 부족하다는 지적을 받고 있다. 초등학교 과정의 경우 소프트웨어 과정은 실과에 포함되어 있으나, 실과 단원과 유기적 관계가 없이 따로 떨어져 구성되어 있으며, 중·고등학교의 경우 정보 교과가 별도로 편성되어 진행되고 있어 다른 교과와의 융·복합 교육이 이루어지기 쉽지 않다. 이러한 문제점을 극복하기 위한 노력으로 최근에는 소프트웨어와 융합인재교육(STEAM)의 연계를 통해 수학, 과학, 미술 등의 교과 중심 융합 소프트웨어 교육 프로그램 개발 및 수업 모형에 대한 연구가 이루어지고 있으나, 대부분 컴퓨터 교육학의 맥락에서 정보 및 실과 교과 운영을 위한 소프트웨어 교육 프로그램의 적용 후 이에 대한 효과성을 검증하거나, 교수학습모형 혹은 수업모형 중심의 연구로

진행되고 있다. 더불어 다수의 연구가 차시별 프로그램이나 교안과 같이 수업의 절차와 방법을 제시한 수업 모형으로서, 교수체제 설계의 개념틀에서 교과-소프트웨어융합을 위한 수업 설계 시 고려해야 할 다양한 요소들에 대한 설명과 전체적인 과정 및 절차를 안내하지 못하고 있다.

본 연구는 교육공학의 맥락에서 국어, 과학, 사회와 같은 기존 교과목의 지식을 소프트웨어와 융합·연계한 교과-소프트웨어융합 수업에 대해 수업설계 모형을 개발하는 것을 목적으로 실행되었다. 이와 같은 목적을 달성하기 위하여 연구문제를 첫째, 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계를 위한 수업설계 원리, 개념모형, 절차모형을 개발하고, 둘째, 개발된 수업설계 모형에 대한 교수자와 학습자의 반응을 검토하는 것으로 설정하였다.

본 연구는 설계·개발 연구방법을 적용하였으며, 다음의 절차로 수행되었다. 우선 선행문헌 고찰을 통해 모형의 구성요소와 설계 원리를 도출한 후, 개념모형과 절차모형을 포함한 초기 모형을 개발하였다. 초기모형의 타당화를 위해 전문가 타당화 검토와 사용성 평가, 현장 평가를 실시하였다. 교수설계 전문가 8인이 두 차례의 전문가 타당화 검토에 참여하였고, 초등학교 교사 3인이 사용성 평가에 참여하였다. 현장 평가에서는 초등학교 담임교사 1인과 해당 학급의 23인의 학생이 참여하였으며, 본 모형이 적용된 수업을 실시하여 학생들의 창의적 문제해결력 및 학업적 자기효능감에 미치는 효과를 질문지를 통해 사전사후 검사를 통해 살펴보았고, 학생 면담을 통해 심층적인 의견을 확인하였다.

연구결과, 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계를 안내하는 개념모형, 절차 모형 및 12개의 설계원리와 33개의 상세지침이 개발되었다. 개념모형은 '교과-소프트웨어융합 교과과정', '교과-소프트웨어융합 수업활동', '교과-소프트웨어융합 학습

지원', '공유 및 평가', '교과-소프트웨어융합 학습도구'의 다섯 가지 구성요소로 이루어져 있으며, 절차모형은 '분석', '설계 및 개발', '실행 및 평가'의 과정에 따라 12개의 세부적인 수업 설계 절차가 포함되었다. 현장 적용 결과, 본 모형을 적용한 수업이 학습자들의 창의적 문제해결력에 긍정적인 영향을 미친다는 것을 유의미하게 도출하였으며, 교사가 창의적 문제해결력을 위한 교과-소프트웨어융합 수업을 체계적으로 설계할 수 있도록 안내하는 효과적인 모형임을 확인하였다.

이상의 연구결과에 기초하여 교과-소프트웨어융합 수업설계 모형, 교과-소프트웨어융합 수업의 지원, 연구의 이론적, 실천적 함의에 대해 논의하였다. 후속연구로, 교과-소프트웨어융합 수업설계 모형을 바탕으로 설계한 수업이 영향을 미치는 요인을 다양하게 확인할 수 있는 연구, 다양한 성찰 방법을 적용한 연구, 학습환경 및 인프라가 부족한 환경에서 효과적으로 수업을 운영하기 위한 교수·학습 전략 및 효과적인 수업 환경 구축, 학교 및 기관의 협력을 통한 시스템 마련, 예비교사에 대한 교과-소프트웨어융합 수업 역량 강화를 위한 프로그램 개발 등에 대한 추가적인 연구의 필요성이 제안되었다.

**주요어** : 교과-소프트웨어융합 수업, 창의적 문제해결, 수업설계 모형  
**학 번** : 2016-32850

# 목 차

I. 서론 .....	1
1. 연구의 필요성 및 목적 .....	1
2. 연구 문제 .....	9
3. 연구의 의의 .....	10
4. 용어의 정의 .....	11
II. 선행문헌 고찰 .....	14
1. 창의적 문제해결과 모형 .....	15
가. 창의성과 창의적 문제해결 .....	15
나. 창의적 문제해결의 특징과 모형 .....	18
2. 컴퓨팅 사고력과 소프트웨어 교육 .....	24
가. 컴퓨팅 사고력과 소프트웨어 교육 .....	24
나. 창의적 문제해결을 위한 소프트웨어 교육의 수업과 모형 .....	35
3. 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과- 소프트웨어융합 수업 설계 .....	41
가. 융합의 정의와 융합인재교육의 특징 .....	41
나. 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과- 소프트웨어융합 교육 .....	51
III. 연구 방법 .....	55
1. 내적 타당화 .....	59
가. 선행문헌 검토를 통한 초기 모형 개발 .....	59
나. 모형에 대한 전문가 타당화 .....	61
다. 사용성 평가 .....	67

2. 외적 타당화 .....	69
가. 연구 참여자 .....	69
나. 현장적용 절차 및 자료 분석 방법 .....	70
다. 검사 도구 .....	72
<b>IV. 연구 결과 .....</b>	<b>75</b>
<b>1. 초기 모형 개발 .....</b>	<b>75</b>
가. 모형의 구성요소 및 설계원리 도출 .....	75
나. 초기 개념모형 개발 .....	89
다. 초기 절차모형 개발 .....	91
<b>2. 내적 타당화 .....</b>	<b>97</b>
가. 구성요소에 관한 전문가 타당화 .....	98
나. 설계 원리에 관한 전문가 타당화 .....	103
다. 개념모형에 대한 전문가 타당화 .....	140
라. 절차모형에 대한 전문가 타당화 .....	147
마. 사용성 평가 .....	153
<b>3. 외적 타당화 .....</b>	<b>159</b>
가. 수업의 설계 및 실행 .....	159
나. 교수자 반응 평가 .....	167
다. 학습자 반응 평가 .....	169
<b>4. 최종 모형 .....</b>	<b>185</b>
가. 개념모형 및 구성요소, 설계원리 .....	187
나. 절차모형의 단계별 설명 .....	208
<b>V. 논의 및 결론 .....</b>	<b>228</b>
<b>1. 논의 .....</b>	<b>228</b>
가. 교과-소프트웨어융합 수업 설계모형 .....	228
나. 교과-소프트웨어융합 수업의 지원 .....	232

다. 연구의 이론적, 실천적 함의 .....	235
<b>2. 결론 및 제언 .....</b>	<b>237</b>
가. 결론 .....	237
나. 연구의 제한점 및 추후 연구를 위한 제언 .....	239
<b>참고문헌 .....</b>	<b>242</b>
<b>부록 .....</b>	<b>261</b>
<b>Abstract .....</b>	<b>303</b>

## 표 목 차

<표 II-1> 다양한 학자들이 제안하는 창의적 문제해결 모형 과 단계 .....	21
<표 II-2> 컴퓨팅 사고의 다양한 정의 .....	25
<표 II-3> 컴퓨팅 사고력의 주요 개념 .....	27
<표 II-4> 컴퓨팅 사고력의 구성요소 비교 .....	28
<표 II-5> MIT Creative Computing 에서의 컴퓨팅 사고력 구성요소 .....	30
<표 II-6> 2015 개정 교육과정 주요 개편 사항 .....	32
<표 II-7> 학교 급별 소프트웨어 교육 목표 .....	33
<표 II-8> 5가지 소프트웨어 교육 교수학습 모델 .....	35
<표 II-9> CT-CPS 수업모형 .....	38
<표 II-10> 창의 컴퓨팅 교수학습 모형 .....	40
<표 III-1> 설계·개발 연구의 대표적인 두 가지 유형 .....	56
<표 III-2> 본 연구의 단계별 주요 연구방법 .....	57
<표 III-3> 전문가 패널 프로필 및 타당화 참여 단계 .....	62
<표 III-4> 모형 개발 과정에 대한 전문가 타당화 문항 ..	63
<표 III-5> 구성요소에 대한 전문가 타당화 평가 문항 ..	64
<표 III-6> 개념모형에 대한 타당화 평가문항 .....	65
<표 III-7> 사용성 평가 참여자 프로필 .....	67
<표 III-8> 현장평가 연구 참여자 .....	70
<표 IV-1> 선행문헌 검토를 통한 일반적인 원리 확인 및 구성요소 도출 .....	76
<표 IV-2> 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계를 위한 초기 설계원리 및 상세 지침	83
<표 IV-3> 주요 설계모형의 특성 및 시사점의 절차별 유목화	

.....	92
<표 IV-4> 내적 타당화를 위한 주요 내용 요약 .....	97
<표 IV-5> 구성요소에 대한 1차 전문가 타당화 결과 .....	98
<표 IV-6> 1차 전문가 타당화를 통해 수정된 2차 모형의 구성요소 .....	100
<표 IV-7> 구성요소에 대한 2차 전문가 타당화 결과 .....	101
<표 IV-8> 2차 전문가 타당화를 통해 수정된 3차 모형의 구성요소 .....	102
<표 IV-9> 설계원리 전반에 대한 전문가 타당화 결과 .....	103
<표 IV-10> 초기 개별 설계원리에 대한 1차 전문가 타당화 결과 .....	105
<표 IV-11> 1차 전문가 타당화 검토 의견과 수정 및 개선 사항 .....	106
<표 IV-12> 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계원리와 상세지침 - 2차 .....	108
<표 IV-13> 2차 개별 설계원리에 대한 2차 전문가 타당화 결과 .....	121
<표 IV-14> 2차 전문가 타당화 검토 의견과 수정 및 개선 사항 .....	122
<표 IV-15> 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계원리와 상세지침 - 3차 .....	124
<표 IV-16> 초기 개념모형에 대한 1차 전문가 타당화 결과 .....	140
<표 IV-17> 2차 개념모형에 대한 2차 전문가 타당화 결과 .....	144
<표 IV-18> 초기 절차모형에 대한 1차 전문가 타당화 결과 .....	147

<표 IV-19> 2차 절차모형에 대한 2차 전문가 타당화 결과 .....	150
<표 IV-20> 개념모형, 절차모형, 설계원리에 대한 사용성 평가 결과 .....	154
<표 IV-21> 사용성 평가에서 제안된 개념모형, 절차모형, 설계원리의 강점, 약점, 개선점 및 추가의견	155
<표 IV-22> 개별 설계원리에 대한 사용성 평가 결과 .....	157
<표 IV-23> 모형을 적용한 수업설계 개요 .....	160
<표 IV-24> 창의적 문제해결력, 학업적 자기효능감 사전·사후 검사 결과 .....	170
<표 IV-25> 창의적 문제해결력, 학업적 자기효능감 대응표본 t-검정 결과 .....	170
<표 IV-26> 모형의 가정 .....	185
<표 IV-27> 최종 모형에 반영된 수정 사항 .....	186
<표 IV-28> 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계 원리 및 상세지침(최종) .....	192
<표 IV-29> 학습 맥락과 관련이 있는 비구조화되고 실제적인 문제 설계 시 고려해야 할 기준 .....	213
<표 IV-30> 온라인 학습 플랫폼의 종류 .....	215
<표 IV-31> 소프트웨어학습도구 및 소프트웨어학습지원 웹사이트 .....	217
<표 IV-32> 프로그래밍(Scratch) 코드 평가 항목 .....	218
<표 IV-33> 학습자의 추상화 활동에 대한 질문 목록 .....	223
<표 IV-34> 학습자의 문제해결(자동화) 활동에 대한 질문 목록 .....	224

## 그 립 목 차

[그림 II-1] 선행문헌 고찰 영역 .....	14
[그림 II-2] 창의성의 네 가지 요소 .....	15
[그림 II-3] CPS 6세대 모형 .....	22
[그림 II-4] S3CPS에서의 학습자 활동 모습 .....	24
[그림 II-5] Barefoot Computing 에서의 컴퓨팅 사고력 구성요소 .....	30
[그림 II-6] 소프트웨어 교육의 인재상 .....	32
[그림 II-7] 창의적 문제 해결 모형, 창의성 교육, 프로그래밍 교육의 연관성 .....	37
[그림 II-8] 창의컴퓨팅의 원리 .....	39
[그림 II-9] 학문간 통합 방식 .....	43
[그림 II-10] STEAM 피라미드 .....	45
[그림 II-11] STEAM 교육 수업구성 원리 .....	47
[그림 II-12] STEM 통합 접근의 교육 프로그램 개념 모형 .....	48
[그림 II-13] 활동중심 STEM 교육 프로그램 개발을 위한 절차모형 .....	48
[그림 II-14] STEAM교육 프로그램 개발을 위한 PDIE모형 .....	49
[그림 II-15] PIRPOSAL 모형 .....	50
[그림 II-16] CT-STEAM 수업모형 .....	52
[그림 II-17] 로봇 활용 소프트웨어교육 프로그램의 구조	53
[그림 III-1] 본 연구의 단계별 연구 방법, 진행절차 및 산출물 .....	58

[그림 III-2] 선행문헌 검토를 통한 설계원리 및 모형 도출 과정 .....	60
[그림 IV-1] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 초기 개념모형 .....	90
[그림 IV-2] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 초기 절차모형 .....	94
[그림 IV-3] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 2차 개념 모형 .....	143
[그림 IV-4] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 3차 개념 모형 .....	146
[그림 IV-5] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 2차 절차 모형 .....	149
[그림 IV-6] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 3차 절차 모형 .....	152
[그림 IV-7] 수업 모형과 원리를 바탕으로 작성한 교수·학습 설계안 .....	153
[그림 IV-8] 아이디어 생성을 위한 패들렛 화면 .....	163
[그림 IV-9] 모듈별 스토리보드 작성 장면 및 스토리보드 산출물 .....	164
[그림 IV-10] 모듈별 프로그래밍 제작 활동 및 산출물 공유 장면 .....	165
[그림 IV-11] 모듈별 산출물 발표 장면 및 학습자 간 평가 댓글 화면 .....	166
[그림 IV-12] 생태도시와의 공통점과 차이점을 바탕으로 엔트리를 활용한 홍보자료 아이디어를 도출하는 실제 장면 예시 .....	171
[그림 IV-13] 홍보자료 컨셉에 대한 아이디어 도출 및 학습자간	

피드백 .....	172
[그림 IV-14] 스토리보드 산출물 .....	174
[그림 IV-15] 환경보호를 위한 생태도시 홍보 학습자 산출물 .....	175
[그림 IV-16] 학습자들의 댓글 및 마음(좋아요) 현황 .....	176
[그림 IV-17] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 개념 모형(최종) .....	188
[그림 IV-18] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 절차 모형(최종) .....	210

# I. 서론

## 1. 연구의 필요성 및 목적

4차 산업혁명과 과학기술의 발달에 따라 기존과 다른 새로운 역량이 요구되고 있다. 세계경제포럼(2018)에서 발표된 ‘일자리의 미래’ 보고서에서는 인공지능으로 인해 전문직업의 영역까지 대체하는 시기가 올 것이며, 이에 따라 향후 2022년 사이에 약 7500만개의 일자리가 사라질 것이라고 예측하였다. 하지만 이러한 일자리 감소와 함께 컴퓨터 인공지능 알고리즘이 발전하면서 약 1억 3300만개의 새 일자리를 창출할 것으로 전망하며 미래에는 신산업을 이해하고 활용할 수 있는 기술과 더불어 창의성, 문제해결력, 비판적 사고와 같은 ‘인간’ 영역의 기술에 대한 가치가 증가할 것이고, 이를 갖춘 인재 양성 교육이 필요할 것이라고 예측하였다.

창의성을 갖춘 사람은 풍부한 아이디어를 지니고 있으며 그 지식 기반이 방대하고 새로운 아이디어와 경험에 대해 개방적이다(임철일, 김성욱, 한형중, 서승일, 2014). 일반적으로 창의성(creativity)은 새롭고 적절한 산물을 생산해 낼 수 있는 능력으로 정의된다(신종호, 김민성, 최지은, 허유성, 이지은, 2015). 여기서 산물은 모든 종류의 아이디어와 생산물을 의미하며, “산물이 새롭다”라고 하는 것은 기존의 것을 모방하지 않은 독창적인 면을 가져야 한다는 것이다(신종호 외, 2015). 빠른 속도로 변화하고 있는 미래의 예측 불가능한 테크놀로지 기반 환경에서 미래를 예측하고 현재상황의 문제를 인식하여 새롭고 독창적인 산출물을 내는 창의성은 교육에서 핵심적으로 다루어야 할 중요한 부분이다(전윤식, 김정섭, 윤경미, 2003). 특히 과학 기술의 발전 속도가 가속화될수록 창의성이 경제적 가치 창출의 원동력이 되기 때문에, 창의적 인재 양성이 선진국들의 교육 및 경제 정책의 화두가 되고 있다(최상덕, 2011).

이러한 창의성 계발을 위해 유럽연합에서는 2009년을 “창의성과 혁신

의 해”로 선포하고, 미국에서는 1990년대 말부터 21세기에 요구되는 역량(21st Century Skills) 계발을 위한 교육 개혁을 적극 추진하고 있다. 국제연합(UN)은 2010년에 창조산업 분석을 통해 창의성, 지식, 문화, 기술이 일자리 창출, 혁신, 사회통합을 주도하게 됨을 보여주었다(최상덕, 2011). 국내에서는 2015년 교육부의 개정 교육과정을 통해 폭넓은 기초 지식을 바탕으로 다양한 전문 분야의 지식, 기술, 경험을 융합적으로 활용하여 새로운 것을 창출하는 창의적 사고 역량을 핵심 역량으로 강조하며 미래사회가 요구하는 창의·융합형 인재를 양성하는 것이 목적임을 밝혔다(교육부, 2015a)

이렇듯 미래사회의 핵심역량으로 강조되는 창의성을 결정짓는 다양한 속성 중에서도 창의적 문제해결은 가장 중요한 요소로 관심을 끌며 그동안 꾸준히 연구되어 왔다(송해덕, 2007). 창의적 문제해결은 “이전의 경험에서 얻은 해결구조 또는 해결원리를 현재의 문제에 창의적으로 적용하는 것으로, 일상생활에서 부딪히는 작은 일에서부터 고도의 기술을 필요로 하는 일까지 다양한 상황에서 사용할 수 있는 체계적인 문제해결 방법”을 의미한다(전성균, 서영민, 이영준, 2010). 창의적 문제해결력을 지닌 학습자의 경우 해결책을 구하기 어려운 복잡한 문제들에 직면했을 때, 문제 상황의 핵심요인을 파악하고 새로운 대안을 모색함으로써 효과적인 해결책을 발견할 수 있다는 점에서 의의를 지닌다(송해덕, 2007).

이러한 창의적 문제해결은 최근 중요성이 강조되고 있는 소프트웨어 교육의 주요한 목표로서 다루어지고 있다. 교육부(2015a)는 소프트웨어 교육을 통해 일상생활에서 마주하게 되는 다양한 문제를 이해하고 구조화하여 문제해결 절차에 따라 알고리즘을 설계하고 이를 프로그램으로 구현할 수 있는 컴퓨팅 사고력을 바탕으로 창의융합 인재를 육성하는 것이 목표임을 밝혔다. 영국에서도 소프트웨어 교육의 목표를 정보적 사고, 컴퓨팅 사고력 습득을 통해 세상의 문제를 그 관점으로 보게하고 새로운 것을 창조(make, create)할 수 있게 하는 디지털 창조자(maker, creator) 역량을 강조하며 소프트웨어 교육 과정이 창의성을 바탕으로 문제를 해결하고 새로운 산출물을 만들어가는 것을 목표로 하고 있음을 밝혔다(김

현철, 2015).

이와 같이 국내외 교육과정에 소프트웨어 교육이 반영되고 창의적 문제해결력에 대한 중요성이 강조되고 있음에 따라 소프트웨어 교육을 통한 창의적 문제해결에 대한 연구도 중점적으로 이루어지고 있다. 노지예와 이정민은(2018)는 로봇 활용 소프트웨어 교육이 초등학생의 컴퓨팅 사고력, 창의성, 학업흥미에 유의미한 영향을 주었다고 주장하였고, 이은아(2018)는 창의적 체험활동에서의 코딩교육이 일반계 고등학생의 창의적 문제해결력에 도움이 되었음을 밝혔으며, 김경규와 이종연(2016)은 컴퓨팅 사고력 기반 프로그래밍 학습이 논리적 사고력과 창의적 문제해결능력 향상에 긍정적인 변화를 주고 있다고 하였다.

한 편, 창의적 문제해결력의 신장을 위해 갖추어야 할 미래사회의 새로운 역량으로 컴퓨팅 사고력이 새롭게 떠오르고 있다. 컴퓨팅 사고력을 바탕으로 한 소프트웨어 교육은 문제를 해결하는 과정에서 확산적 사고가 요구되며, 그 중 최적의 해를 탐구하고 적용하는 과정에서 수렴적 사고가 요구되므로, 수렴적 사고와 발산적 사고를 토대로 이루어지는 창의적 문제해결력 신장에 긍정적인 영향을 미친다(김경규, 이종연, 2016; 류미영, 한선관, 2015; 전용주, 2017; 한영신, 2018). 컴퓨팅 사고력은 미래사회를 살아갈 수 있는 핵심적인 역량으로서 추상화(abstraction)와 자동화(automation)를 통해 복잡한 시스템을 설계하거나 어려운 문제를 해결하는 것이다(Wing, 2006). 컴퓨팅 사고력은 인간이나 기계가 효과적으로 수행할 수 있는 방식으로 문제를 공식화하고 해결책을 표현하는 데 수반되는 사고 과정으로, 읽기, 쓰기, 셈하기와 같이 누구나 갖추어야 하는 기본적인 역량으로 인식되고 있다(Wing, 2006). Wing은 단순히 컴퓨터 과학 및 프로그래밍에 대한 지식을 가지고 있다거나 프로그래밍을 실천하는 기술적인 능력을 가지고 있는 것만이 컴퓨팅 사고력을 가지고 있는 것이 아니라, 문제의 인식, 자료 수집과 분석, 문제 해결책 마련을 위하여 다양한 사고를 진행하고, 그 과정에서 컴퓨팅 능력을 활용하고 그 해결책을 컴퓨팅적으로 구현하는 방법을 설계하는 과정 전반을 컴퓨팅 사고력으로 생각하고 있다(이영준 외, 2014; Wing, 2006; Wing, 2011).

컴퓨팅 사고력을 바탕으로 창의적 문제해결력을 신장하기 위한 소프트웨어 교육의 수행을 위해 전용주(2017)는 컴퓨팅 사고력 기반 창의적 문제해결(CT-CPS)수업모형을 개발하였다. CT-CPS 수업모형은 문제의 인식 및 분석 단계에서 자료 수집, 분석, 표현에 대한 다양한 활동을 진행하면서, 스스로 문제를 찾아 컴퓨팅 사고력을 통해 해결하도록 한다는 점에서 기존 소프트웨어 교수학습모형과의 차별성을 가진다(전용주, 2017). 이 후 구재훈과 김태영(2018)은 초등 정보과학 영재를 대상으로 창의적 문제해결력 향상을 위한 융합형 CT-CPS프레임워크 기반 수업 콘텐츠 설계 연구를 수행하였으며, 구재훈, 전용주, 김태영(2016)은 초등 정보영재의 창의적 문제해결력 향상을 위해 CT-CPS프레임워크 기반 수업 콘텐츠를 개발하여 적용한 결과 창의적 문제해결력의 향상도가 유의미한 것을 밝혔다.

이와 같이 컴퓨팅 사고력은 일상적인 생활의 문제를 창의적으로 해결해나갈 수 있는 중요한 역량이자 창의적 문제해결력을 신장시킬 수 있는 사고과정이다. 컴퓨팅 사고력과 이를 통한 창의적 문제해결력 등 미래 사회의 새로운 역량이 강조되는 추세에 따라, 영국, 미국 등 주요 국가에서는 컴퓨터 과학을 정규 교육과정으로 지정하여 학생들의 컴퓨팅 사고력을 통한 창의적 문제해결력을 증진시키기 위한 노력이 최근 몇 년 사이에 이루어지고 있다. 영국은 2013년 9월 소프트웨어 교육을 위한 교육과정을 공시하고 2014년 9월부터 본격적으로 시행하고 있으며, 교육과정 개편을 통해 12개 정규 과목 중 하나로 포함된 컴퓨터과학은 영어, 수학, 과학 등과 더불어 5대 필수 과목으로 지정되었다(송기상, 소효정, 2017; 우은경, 2019). 미국은 연방국가의 특성상 국가 단위의 소프트웨어 교육 과정이 재정되지는 않았지만, 백악관을 통한 소프트웨어 교육 캠페인을 비롯해 code.org 등 민간기관에서의 활발한 소프트웨어 교육운동이 일어나고 있다(The White House, 2016; Scott, 2016).

국내의 경우 2015년 개정교육과정을 바탕으로 2018년도부터 시행된 소프트웨어 교육은 초등학교의 경우 실과 교과에 반영되어 연간 17차시 이상을 확보하도록 하였고, 중학교의 경우 ‘정보’과목을 통해 연간 34차

시 이상을 시행하도록 하였으며, 고등학교는 ‘기술·가정’교과군의 심화 선택과목이었던 ‘정보’교과를 일반 과목으로 변경하였다(교육부, 2015a). 이전 교육과정과 비교하였을 때 가장 큰 차이는 컴퓨팅 사고력을 핵심 사고전략으로 두어 문제해결중심의 사고와 추론의 능력을 기르도록 하였다는 점이다(송기상, 소효정, 2017). 즉, 소프트웨어 교육을 통해 일상생활에서 마주하게 되는 다양한 문제를 이해하고 구조화하여 문제해결 절차에 따라 알고리즘을 설계하고 이를 프로그램으로 구현할 수 있는 컴퓨팅 사고력을 신장하는 것이 미래 사회의 창의적 융·복합인재 육성을 위한 개정 교육과정의 목표임을 알 수 있다(교육부, 2015a)

하지만 2015년 새롭게 개정된 교육과정과 소프트웨어 교육 자체에 대한 많은 장점에도 불구하고, 직접적인 소프트웨어 교육의 시수는 다른 외국 주요 국가에 비해 부족한 실정이다. 현재 국내 개정 교육과정에서 지정하고 있는 소프트웨어 교육 시수는 초등학교 17시간, 중학교 34시간으로, 외국 주요 국가에서 지정하고 있는 소프트웨어교육 시수와 비교하였을 때 큰 차이를 보이고 있다. 특히 영국의 경우 초등학교 180시간, 중학교와 고등학교 90시간을 필수로 지정하고 있으며, 중국은 초등학교 3학년부터 연간 70시간 이상, 일본은 중학교 55시간, 고등학교 70시간, 인도는 초·중·고등학교 모두 180시간으로 의무화 하였다. 이는 초등학교 17시간, 중학교 34시간으로 소프트웨어 교육 시수를 지정하고 있는 국내 교육과정에 많은 차이가 나고 있는 상황으로, 추 후 시수를 점차 확대해 가거나, 수학, 과학 등 기존 교육과정에 소프트웨어 교육을 활용하는 등 해결책을 마련할 필요성이 있다(김현철, 김서영, 이현정, 김영일, 한건우, 2018).

특히 현재 각 학교급에서 이루어지고 있는 소프트웨어 교육은 다른 교과와의 연계성이 부족하다는 지적을 받고 있다. 초등학교 과정의 실과 과목의 경우 실과와 따로 떨어진 ‘하나의 섬’처럼 구성되어 다른 교과와의 유기적인 관련성이 부족하다(정영식, 2017). 중·고등학교의 경우 정보교과가 별도로 편성되어 진행되고 있어 다른 교과와의 융·복합교육이 이루어지기는 쉽지 않을 수 있으며, 소프트웨어교육과정에서 타교과 부분

까지 컴퓨팅 사고 반영을 언급하는 것은 교과 월권행위로 비취질 수 있고, 현실적으로 실현 가능성이 낮다는 지적도 받고 있다(이철현, 2015).

이상에서 언급되고 있는 것처럼, 컴퓨팅 사고력을 바탕으로 창의적 문제해결력을 신장시키기 위한 소프트웨어 교육은 미래사회를 대비한 새로운 시도와 장점 및 가능성에도 불구하고, 과학, 수학 등 기존 과목과의 유기적인 교과융합 학습을 설계하고 수업에 적용하기는 것이 쉽지 않은 과제이다. 여기서 ‘융합’은 서로 다른 이질적인 여러 요소들이 일정한 원리나 절차에 따라 의도적으로 결합하여 새로운 하나의 통일체를 형성하는 것을 의미하며, 학문 깊이에서의 질적 확대와 학문 영역들 간의 시너지를 창출한다는 의미이다(“융합” in 국립국어원 표준국어대사전; 이운발, 2005). 교과융합의 대표적인 형태인 융합인재교육(STEAM)또한 학문 간의 융합을 통해 다양한 교과 영역을 자연스럽게 습득하면서 창의적이고 색다른 결과를 도출하며 학문 영역들 간의 시너지를 창출하는 데 목적이 있다(이운발, 2005)

따라서 국어, 과학, 수학 및 융합교과(STEAM)와 관련된 문제들을 컴퓨팅적 사고력으로 접근하고 해결하기 위해서는 다양한 문제를 컴퓨팅 사고력을 활용하여 바라보고 창의적인 해결책을 모색할 수 있는 훈련이나 경험이 필요하다. 마치 ICT를 전체 교과목의 교수·학습 활동에 활용하는 것처럼, 소프트웨어 교육도 전 교과목들과 유기적 관련성을 갖도록 편성·운영될 필요가 있으며(임정훈, 2018), 교과융합 소프트웨어 교육이 효과적으로 실행되기 위해서는 교수설계 측면에서 세부적으로 고려해야 할 부분이 많다.

기존 과목에 소프트웨어 교육을 융합한 교과융합 소프트웨어 교육의 필요성이 증대되고 있음에 따라, 다양한 과목과 융합한 소프트웨어 교육에 대한 연구가 이루어지고 있다. Ke(2014)는 중학교 학생을 대상으로 블록형 코딩 프로그램을 활용하여 수학 게임을 제작하는 연구를 진행하였으며, 이를 통해 학습자는 수학에 대한 자아효능감, 가치, 즐거움, 동기를 증진하였고 추상적, 정량적인 추론 능력이 강화되었다고 보고하였다. 또한 수학, 영어, 사회, 과학과 같은 기존 교과목과의 소프트웨어 융합교

육은 해당 교과에 대한 이해와 성찰(Burke, 2012), 컴퓨팅 사고력, 창의성(서영호, 엄미령, 2016), 문제해결력(Calao, Moreno-León, Correa, & Robles, 2015; 김명인, 2018), 동기(Wolz et al., 2011) 향상에 도움이 되었다. 소프트웨어 융합 교과의 프로그램 및 커리큘럼, 교수·학습모형 개발에 대한 연구도 진행되고 있다. 특히 국내의 경우 초등학생·중학생 대상 수학, 과학, 미술 등의 교과 중심 융합 소프트웨어 교육 프로그램 개발 연구와 교수·학습 모형 개발 및 적용 연구 등 프로그램 개발 및 교수·학습 모형에 대한 연구가 활발히 이루어지고 있다(류미영, 한선관, 2015; 이미화, 함성봉, 2018; 이혜영, 이태욱, 2015). 또한 컴퓨팅 사고력과 융합교과(STEAM)의 연계를 위해 미국의 일부 대학과 교육 단체에서는 컴퓨팅 사고력과 STEM을 융합한 형태인 C-STEM에 대한 지원이 활발히 시행되고 있다. C-STEM은 STEM 교과목에 프로그래밍과 피지컬 컴퓨팅 등의 소프트웨어 교육을 융합시킨 소프트웨어 융합인재교육으로 STEM 과목과 코딩 및 소프트웨어를 융합한 여러 가지 교육 프로그램과 교사 교육 등이 제공되고 있다.

하지만 위와 같은 소프트웨어 융합 교육에 대한 사례 및 연구는 대부분 컴퓨터교육학의 맥락에서 정보 및 실과 교과 운영을 위한 소프트웨어 교육 프로그램의 적용 후 효과성을 검증하거나, 교수·학습모형 혹은 수업모형 중심의 연구로 진행되고 있다. 최근에 주로 이루어지고 있는 소프트웨어 교육 연구에서의 교수·학습 모형 연구는 정보·실과 교과목에 한정된 소프트웨어 수업을 위한 수업 모형으로서, 과학, 수학, 융합교과(STEAM)과 같은 기존 과목을 중심으로 이루어지는 융합교육 맥락의 소프트웨어 수업에 대한 설계가 부족하다. 더불어 다수의 연구가 차시별 프로그램이나 커리큘럼, 교안과 같이 수업의 절차와 방법을 제시한 수업 모형으로서, 교수체제설계(Instructional Systems Design)의 개념틀에서 교과융합을 위한 수업 설계 시 핵심적으로 고려해야 할 다양한 요소들에 대한 설명이 부족하고 전체적인 과정과 절차를 안내하지 못하고 있다(김성욱, 2016; 임정훈, 2018).

이에 본 연구에서는 창의적 문제해결력의 개발을 목적으로 초등학교

맥락에서 사회 혹은 과학과 같은 기존 교과목의 지식과 융합·연계한 컴퓨팅 사고력 기반 교과-소프트웨어융합 교육에 대해 수업설계 모형을 개발하고자 한다. 본 연구를 통해 개발되는 교과-소프트웨어융합 수업설계모형은 초등학교 맥락의 학교 현장에서 수업을 이끌어 나가는 교사가 이 설계모형을 통해 보다 쉽고 효과적이게 교과-소프트웨어융합 교육을 설계하여 궁극적으로 학습자의 창의적 문제해결력을 개발할 수 있도록 하기 위함이다. 즉, 교육공학의 맥락을 바탕으로 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 안내하는 수업설계 모형을 개발하는 것이 본 연구의 목적이다. 이를 위해 창의성, 창의적 문제해결, 컴퓨팅 사고력, 소프트웨어 교육, 융합교육, 교과 융합 소프트웨어 교육, 수업 설계와 관련된 선행문헌을 고찰하여 초등학교 맥락의 교과-소프트웨어융합 수업설계 원리 및 개념 모형, 절차 모형을 개발하고, 관련 분야의 전문가 검토와 교사의 사용성 평가를 바탕으로 수업설계 원리와 모형의 타당성을 검토하고 개선 및 보완하였다. 이 후 개발된 수업설계 원리와 모형을 실제 초등학교 수업 현장에 적용하고 교사와 학습자의 반응을 분석하여 최종 모형을 도출하였다.

## 2. 연구문제

본 연구는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계모형을 개발하는 것을 목적으로 한다. 이를 위해 본 연구에서 설정한 연구문제는 다음과 같다.

1. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계모형은 무엇인가?

1-1. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계원리는 무엇인가?

1-2. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 개념모형은 무엇인가?

1-2. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 절차모형은 무엇인가?

2. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계모형은 타당한가?

2-1. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계모형은 내적으로 타당한가?

2-2. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계모형은 효과가 있는가?

### 3. 연구의 의의

본 연구의 목적은 교육공학의 맥락을 바탕으로 교과-소프트웨어융합 수업을 안내하는 수업 설계모형을 개발하고 이를 타당화 하는 것이다. 본 수업설계 모형은 초등학교 맥락에서 교과-소프트웨어융합 수업 설계 환경과 컴퓨팅 사고력 기반의 창의적 문제해결 설계 활동을 위한 수업설계의 모형과 원리를 제시하였다. 본 연구의 실천적·이론적 의의는 다음과 같다.

첫째, 창의적 문제해결력 증진을 목표로 교수설계자가 다양한 방식의 교과-소프트웨어융합 수업 설계에 참고할 만한 수업설계 원리와 모형을 제공한다. 기존의 소프트웨어 수업설계와 관련한 연구는 실과나 정보 교과 맥락에서 교육 프로그램을 설계하고 이에 대한 효과성을 검증하는 연구 위주로 이루어져 구체적으로 수업설계의 원리와 절차를 안내하는 연구가 부족한 상황이다. 따라서 본 연구를 통해 교육공학의 맥락을 바탕으로 우리나라의 실제 교육현장에 적용할 수 있는 실천적 모형을 개발하여 교수자에게 필요한 안내와 지원 사항을 제공할 수 있다.

둘째, 타 교과 지식과 소프트웨어 교육의 융합수업 설계를 위한 절차와 활동, 세부 전략을 제공한다. 기존 실과 및 정보 교과가 다른 교과와의 유기적인 관련성이 부족하다는 지적을 받고 있음에 따라(정영식, 2017), 사회, 국어, 과학과 같은 기존 과목과의 교과융합 학습을 설계하고 수업에 적용하기는 쉽지 않은 실정이다. 이에 본 연구에서는 창의적 사고 촉진 및 컴퓨팅 사고력에 기반 한 국어, 사회 및 STEAM과 같은 교과-소프트웨어융합 수업설계원리 및 모형을 개발한다. 이를 통해 효과적인 창의적 문제해결력 향상을 위한 수업설계활동을 위한 시사점을 제공해 줄 것이다.

셋째, 초등학교 맥락에서 이루어지는 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 프로그램 개발에 도움이 될 것이다. 소프트웨어 교육, 창의적 문제해결의 중요성과 역할이 높아지면서 컴퓨팅 사고력 기반 창의적 문제해결 시도가 이어지고 있다(전성균 외, 2010;

경인교육대학교 미래인재연구소, 2015; 전용주, 2017). 특히 초등학교의 소프트웨어 교육 내용은 이후 중학교와 고등학교의 정보 교과로 심화 및 확대되므로 학교급별 내용요소와 계열성을 고려하여 기초적인 컴퓨팅 사고력 및 창의적 문제해결력 증진을 위한 종합적인 교과-소프트웨어융합 수업을 설계하여 제공하는 것이 필요하다. 본 연구에서 개발한 수업설계 원리와 모형은 실제 초등학교의 교과 지식과 소프트웨어를 융합한 수업을 융합한 설계원리로서, 초등학교 맥락에서 창의적 문제해결을 위한 교과-소프트웨어융합 수업 개발에 대한 기초 자료로서 활용될 수 있다.

넷째, 소프트웨어 교육에 대한 수업설계를 시도함으로써 교육공학의 학문적 지평을 확대하는 것에 도움이 될 것이다. 이제까지 소프트웨어 교육은 컴퓨터 공학 및 컴퓨터 교육 분야에서 이루어진 연구와 비교하여 교육공학 분야에서의 연구는 상대적으로 많이 이루어지지 않고 있는 실정이다. 본 연구를 통해 교육공학 분야의 새로운 연구 주제로서 창의적 문제해결을 위한 교과-소프트웨어 교육을 제시하여 교육공학의 학문적 범위 확대에 기여할 수 있다.

## 4. 용어의 정의

### 가. 컴퓨팅 사고력

컴퓨팅 사고력(Computational Thinking)이란 ‘문제를 구성하고 그 문제의 해결책을 제시하는 사고 과정에서 정보처리 체계를 통해 효과적으로 처리될 수 있는 형태로 제시할 수 있는 능력’이다(Wing, 2006). 컴퓨팅 사고력은 Wing(2006)과 ISTE & CSTA(2011)에서 내린 정의를 토대로 한국과학창의재단(2014)을 중심으로 이를 ‘컴퓨팅 사고력’이라는 한국어로 표기하기로 합의하였고, 교육부(2015)에서 ‘컴퓨터 과학의 기본 개념과 원리 및 컴퓨팅 시스템을 활용하여 실생활과 다양한 학문분야의 문제를 이해하고 창의적으로 해법을 구현하여 적용할 수 있는 능력’이라고 재정의 하였다. 컴퓨팅 사고력의 주요 개념은 자료 수집, 자료 분석, 자

료 표현, 문제 분해, 추상화, 알고리즘 및 절차, 자동화, 시뮬레이션, 병렬화로 구분될 수 있다. 즉 컴퓨팅 사고력은 문제의 이해를 토대로 자료를 수집 및 분석하고 논리적으로 배치한 후, 문제의 핵심 내용을 설정하는 추상화를 통하여 자료를 표현하고 절차적 사고력을 통해 해결과정을 자동화 하는 과정을 거쳐, 가장 효과적이고 효율적으로 문제를 달성하기 위한 해결책을 확인, 분석, 실행하고 문제해결 과정을 일반화하고 전이하는 과정을 설명한다. (ISTE & CSTA, 2011). 본 연구에서는 컴퓨팅 사고력을 한국과학창의재단에서 한국어로 표기한 컴퓨팅 사고력과 Wing(2006) 및 교육부(2015a)가 제시한 정의를 토대로 ‘실생활과 다양한 학문분야의 문제를 이해하고 창의적으로 해결책을 구현하여 정보처리 체계를 통해 효과적으로 적용할 수 있는 능력’으로 그 의미를 재정의 하여 사용하였다.

#### 나. 창의적 문제해결

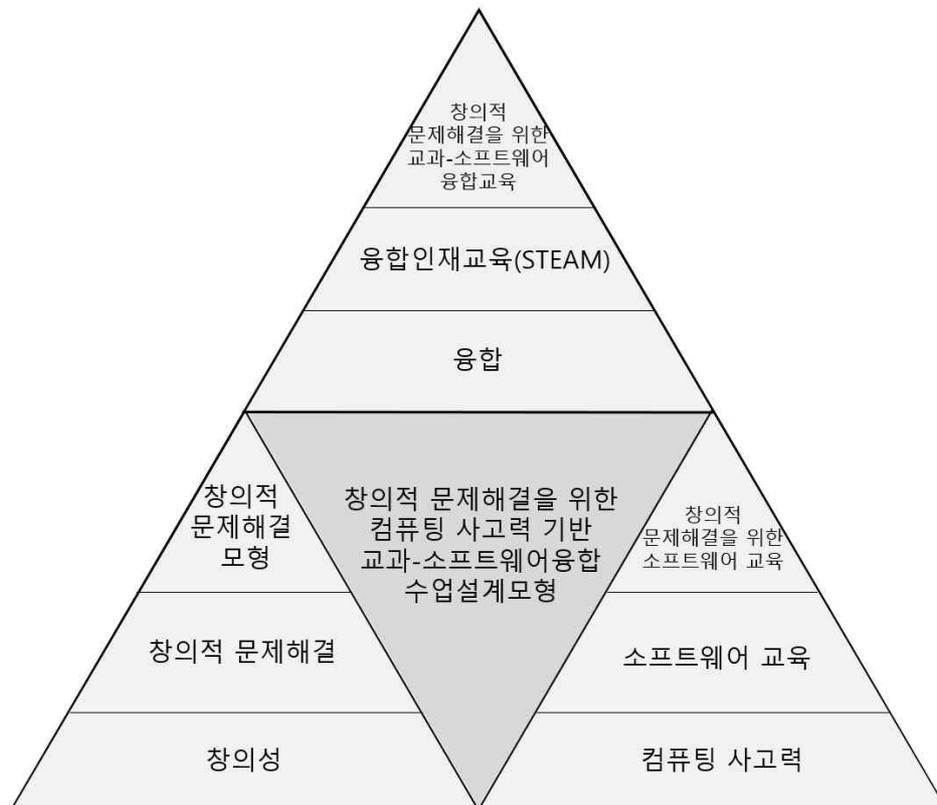
창의적 문제해결은 일상생활에서 직면할 수 있는 애매모호하고 복잡한 문제에 접했을 때, 다양한 해결책이 나올 수 있는 문제를 해결함에 있어서, 여러 대안들 가운데 최선의 것을 찾는 과정이다(김민영, 2006; 황지현, 2010). 창의적 문제해결에서 다루는 문제는 비구조화된 문제이며, 창의적 문제해결은 비구조화된 상태의 문제를 이해하고, 아이디어를 산출한 뒤 이를 실행에 옮길 수 있도록 계획하는 과정에서 확산·수렴적 사고가 반복적으로 작용하는 과정이라고 할 수 있다(Treffinger, Isaksen, & Dorval, 1994; 2000). 따라서 본 연구에서는 이러한 창의적 문제해결을 ‘비구조화된 문제를 해결하는 과정에서 확산·수렴적 사고 통해 다양한 아이디어를 생성하고 문제에 대한 해결책을 찾아내는 체계적인 사고과정’이라는 의미로 사용하였다.

#### 다. 교과-소프트웨어 융합교육

교과-소프트웨어 융합교육은 융합인재교육(STEAM)과 같은 타 교과를 중심으로 소프트웨어 교육을 융합한 것으로, 타 교과 지식과 소프트웨어 교육의 연계를 통해 다양한 분야를 융합적으로 활용하여 새로운 것을 창출하는 창의적 사고 역량을 신장시키기 위한 교육이다(교육부, 2015b; 김혜란, 최선영, 2019a; 배운주, 이정민, 2020). 본 연구에서는 ‘교과교육의 목표를 달성하고 학습자의 창의융합 능력을 향상시키기 위해 교과 활동을 바탕으로 문제의 분석과 해결방안 등을 컴퓨팅 사고력 기반의 소프트웨어 교육과 연계하여 수행하는 것’이라는 의미로 사용하였다.

## II. 선행문헌 고찰

본 연구는 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형을 개발하는 것을 최종적인 목적으로 한다. 연구목적을 달성하기 위하여 다음과 같은 사항을 중심으로 선행문헌을 고찰하였다. 첫째, 교과-소프트웨어 융합 수업설계를 위한 창의성과 창의적 문제해결의 이론적 기저와 창의적 문제해결모형의 구성요소를 파악하였다. 둘째, 컴퓨팅 사고력의 개념 및 구성요소와 소프트웨어 교육의 목적 및 현황, 그리고 창의적 문제해결을 위한 소프트웨어 교육의 수업 모형을 확인하였다. 마지막으로 융합과 융합인재교육(STEAM)의 개념과 특징을 확인하고, 교과-소프트웨어 융합교육에 대해 탐색함으로써 창의적 문제해결을 위한 교과-소프트웨어 융합교육에 대한 이론적 기초를 고찰하였다.



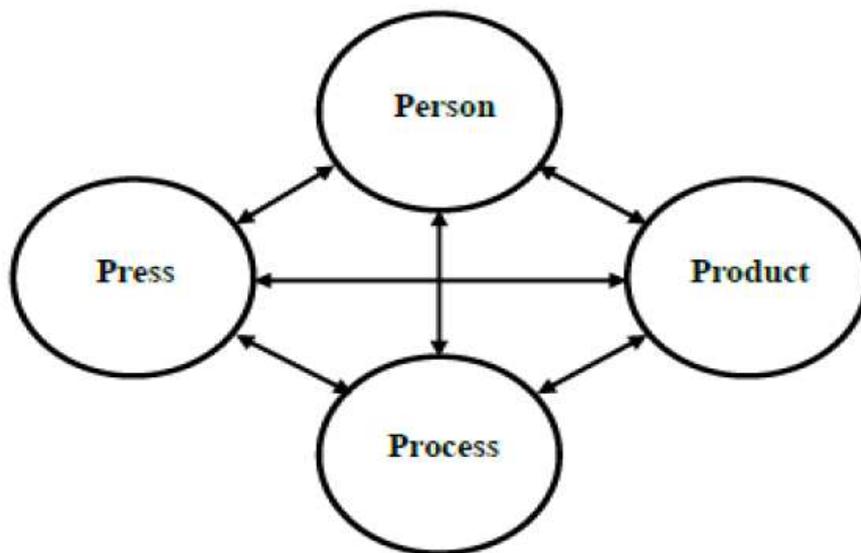
[그림 II-1] 선행문헌 고찰 영역

# 1. 창의적 문제해결과 모형

## 가. 창의성과 창의적 문제해결

### 1) 창의성의 개념과 정의

창의성에 대한 연구는 1950년 미국심리학회(The American Psychological Association) 회장이었던 Guilford가 창의성이라는 주제로 기초연설을 하면서부터 본격적으로 시작되었다. 이후 창의성은 다양한 접근으로 연구되었으며, 여러 학자들에 의해 다양한 정의가 제시되었다. 창의성의 정의를 분류하기 위해 Rhodes(1961)는 창의성에 대한 정의를 수집하고 비교·분석하였으며, 사람(Person), 과정(Process), 산출물(Product), 환경(Press)의 네 가지 요소로 구분하여 정리하였다([그림 II-2]참조).



[그림 II-2] 창의성의 네 가지 요소(The 4Ps of Creativity)  
(Cropley, 2016; Rhodes, 1961)

‘사람’의 관점에서 본 창의성은 개인의 성격이나 특성으로 창의성을

정의한 것으로, ‘창의적 사람’이 어떠한 사람이고 특성을 갖는지에 대한 접근이다(김영채, 1999; 홍미영, 2012). Guilford(1967)는 창의적인 사람은 문제에 대해 하나의 정답을 제시하기보다는 새로운 답이나 아이디어를 많이 생성하는 능력이 있다고 가정하였으며, 확산적 사고를 통해 사람들의 창의적 잠재력을 이해하고자 했다. Torrance(1972)는 창의적인 사람들은 어떤 문제를 해결해 나가거나 아이디어를 만들어 내는 과정에서 전통적으로 생각하는 틀에서 벗어나 특이한 사고를 하는 것을 의미하는 것이라 하였으며, Gardner(1993)는 창의적인 사람은 문제를 해결하고 산출물을 만들거나 새로운 문제를 만들어 내는 사람이라고 정의내리고 있다. 또, MacKinnon(1978)은 보통사람들과는 다른 매우 창의적인 사람들의 ‘개인 특성’을 밝혀서 창의성의 본질을 설명하고자 하였다.

‘과정’은 창의성을 사용할 때 일어나는 사고과정에 대한 접근(김영채, 1999)으로, Rhodes(1961)는 동기 유발, 인식, 학습, 사고, 대화 등과 관련이 있다고 하였다. Mednick(1962)는 창의성에 대해 새로운 생각의 형태로 일상적이지 않은 과정을 포함하는 것이라 설명하고 있다. 조연순, 정혜영, 백은주, 정혜영, 최규리, 임현화(2011)의 연구에서는 이미 알고 있는 해결책을 벗어난 생각들 사이에서 새롭고 연결되기 어려운 것들을 연결시키기 위한 과정의 필요성을 강조하였다. 또한 Torrance(1974)는 창의성을 창의적 과정으로 정의하고, 문제와 정보의 차이와 잘못된 부분을 정의하고, 이에 대한 가설을 설정, 평가, 검증하고, 다시 수정을 거쳐 재검증한 후, 결과를 제시하는 과정을 모두 포괄하는 것이라고 하였다. 또한 ‘과정’의 창의성 연구에서는 창의적인 과정이 진행되는 것을 설명하기 위해 창의적인 문제해결력의 단계를 제시하거나 아이디어를 통합하는데 초점을 둔다. Wallas(1926)는 창의적 과정을 준비 단계, 부화 단계, 영감 단계, 검증 단계의 네 단계로 설명하였고, Osborn(1963)은 창의적 문제해결력 단계를 문제를 ‘사실 발견’, ‘아이디어 발견’, ‘해결안 발견’의 세 가지로 제시하였다.

‘산출물’의 관점에서는 창의성을 창의적인 과정을 통해서 아이디어를 실제적 형태로 바꾼 산출물로 본다(홍미영, 2012). 창의적 산출물은 개인

적인 것일 수도 있고, 집단적인 것일 수도 있으며, 유형과 무형의 산출물을 모두 포괄한다(김영채, 1999). Amabile(1983)은 창의성은 사람의 특성을 나타내는 것이 아니라, 새롭고 적절한 아이디어나 행동 혹은 산출물을 의미한다고 주장하였다. 또한 Urban(1995)은 창의성을 문제를 해결한 결과로 새롭고 놀랄만한 산출물을 만들어 내는 것이라고 정의하여, 창의성을 문제해결 과정의 산출물로 보았다.

마지막으로 ‘환경’은 인간과 환경간의 관계로서, 창의성이 일어나는 장면, 장소, 맥락 또는 분위기를 다룬다(김영채, 1999; Rhodes, 1961). 창의적인 환경 측정에서의 ‘환경’이란 창의적인 사고 과정이 처음으로 자극되어지고, 또한 창의적인 산출물을 완성하기까지의 사고 과정이 유지되도록 도와주는 ‘총체적이고 복잡한 일체의 상황’을 의미한다(Taylor, 1988). 총체적인 환경이란 한 사람이나 여러 사람들의 창의적인 사고 과정을 유지하고, 좀 더 창의적인 사고를 불러일으킬 수 있는 교수 매체까지도 의미한다(전경원, 2006)

위에서 살펴본 바와 같이 창의성 개념은 학자들에 의해서 다양하게 정의되고 있으며, Rhodes(1961)는 이러한 연구들을 종합 및 분석하여 창의성을 4P로 정의하고 있다. 한편 Rhodes(1961)는 창의성을 이해하기 위해서는 사람과 사람, 과정, 산출물, 환경 등을 통합적으로 바라볼 필요가 있다고 주장하였다. 창의적인 산출물이란 창의적인 사람이 창의적인 과정을 통해 얻은 결과물이며, 이런 모든 것들은 창의적인 환경에 의해 지원받을 때 가능한 것이다(Rhodes, 1961). 즉 창의성은 사람, 과정, 산출물, 환경 요소가 상호작용을 할 때 창의성이 발휘되며, 결과적으로 수업 진행 시 학습자가 창의성을 발휘하도록 가르치기 위해서는 복합적인 요소를 고려하여 수업 설계를 할 필요성이 있다.

## 2) 창의성과 창의적 문제해결의 관계

본 연구에서는 창의성을 보는 복합적인 요소 중 ‘과정’ 측면에 집중하여 창의적 문제해결의 과정과 단계를 살펴보고자 한다. 일반적으로 문제

해결 연구와 창의성 연구는 각기 분리되어 진행되어 왔지만, 여러 연구자들이 ‘창의성’과 ‘문제해결’을 매우 밀접한 관계로 관련지어 정의하고 있다(조연순, 성진숙, 이혜주, 2008). Guilford(1967)와 Basadur(1994)는 창의성과 문제해결을 같다고 보았으며, 창의적 사고는 새로운 결과를 이끌어 내는 것이고, 문제해결은 새로운 상황에 대한 새로운 반응을 산출하는 것이므로 기본적으로 같은 정신적 현상으로 볼 수 있다고 주장하였다. 또한 Feldhusen과 Treffinger(1983)는 유창성, 융통성, 독창성 등이 복합적인 문제해결에 꼭 필요한 구성요소임을 주장하면서, 창의성과 문제해결을 ‘하나의 복합적인 개념’으로서 설명하였다.

한편 Newell, Shaw와 Simon(1962)은 창의적 행위를 문제 형성에서의 새로움, 비관습성, 인내심, 어려움으로 특징화되는 문제해결 활동의 한 특별한 종류라고 언급하였고, Mumford, Reiter-Palmon과 Redmond(1994)도 창의적 사고를 문제해결의 한 형태라고 하였다.

이러한 창의적 문제해결에 대해 Treffinger, Isaksen과 Dorval(2000)은 ‘발산적 사고와 수렴적 사고를 반복적으로 활용하여 어떤 상황에 놓인 비구조화된 문제를 발견하고 그 문제와 관련된 자료를 음미하고, 구체적인 문제를 진술함으로써 창의적으로 다양한 아이디어를 생성하고 해결안을 결정하는 사고 과정’이라고 정의한다. 또한 Cropley(2001)는 잘 정의되지 않은 문제해결을 위해서는 먼저 문제를 인식하고 정의해야 하며, 이후 문제를 해결하고, 해결책에 대한 증거를 마련해야 한다고 주장하였다. 이렇게 많은 연구자들은 창의성과 창의적 문제해결을 연관시켜 동일하면서도 복합적인 개념으로 보고 있다.

## 나. 창의적 문제해결의 특징과 모형

창의적 문제해결력에 관해 알기 위해서는 우선 문제가 무엇이고, 문제해결의 개념을 명확하게 정의해야 한다. 김영채(1999)는 문제를 ‘시초상태(현재상태)’와 ‘목표상태’ 사이에 장애가 있어 거리(간격, 괴리)가 있는 것으로 정의하였다. Ernst와 Newell(1969)는 문제의 구성요소로 시초의

상태, 목표상태, 조작인 및 조작인의 제한 등의 네 가지를 제시하였다. 시초의 상태란 문제에 대하여 주어진 정보를 의미하며, 목표상태란 최후의 목표장면에 대한 정보이다. 또한 조작인은 조작자, 조작 행위를 말하는 것으로, 어떤 상태를 다른 상태로 변화시키기 위하여 수행할 수 있는 작용 또는 이동행위(해결방법)을 의미한다. 조작인의 제한이란 조작인의 적용을 구속하고 지배하는 규칙을 말한다. 또한 문제는 구조화의 정도에 따라 '구조화된(well-structured)문제'와 '비구조화된(ill-structured) 문제'로 구분할 수 있다(Frederiksen, 1984; Jonassen, 1997; Simon, 1973). '구조화된 문제'는 정의가 잘 된 문제로서, 문제해결자에게 문제를 해결하는데 필요한 시초의 상태, 목표상태, 조작자, 조작인 제한 등에 대한 정보를 모두 제공한 다음 해결을 요구한다(김영채, 1999). 따라서 구조화되거나 잘 정의된 문제는 분명하게 진술되어있고, 문제해결을 위한 표준화된 절차나 방법이 알려져 있거나 주어진다(조연순 외, 2008). 반면 '비구조화된 문제'는 정의가 잘 되어 있지 않은 문제로서, 시초의 상태, 목표상태, 조작인, 조작인의 제한 등에 관한 정보가 부족하거나 주어지지 않은 경우를 말한다(김영채, 2004; 조연순 외, 2008). 특히 비구조화된 문제는 우리가 일상생활에서 끊임없이 부딪치는 문제로, 문제 자체도 분명하지 않고, 그 해결책 또한 다양한 문제이다.

창의적 문제해결은 일상생활에서 직면할 수 있는 애매모호하고 복잡한 문제에 접했을 때, 다양한 해결책이 나올 수 있는 문제를 해결함에 있어서, 여러 대안들 가운데 최선의 것을 찾는 과정이라 할 수 있다(김민영, 2006; 황지현, 2010). 즉, 창의적 문제해결에서 다루는 문제는 Jonassen(1997)이 말하는 비구조화된 문제이며, 창의적 문제해결은 이러한 비구조화된 상태의 문제를 이해하고, 아이디어를 산출한 뒤 이를 실행에 옮길 수 있도록 계획하는 과정에서 확산·수렴적 사고가 반복적으로 작용하는 과정이라고 말할 수 있다(Treffinger, Isaksen, & Dorval, 2000).

앞서 언급했듯이 창의적 문제해결에 대한 정의는 여러 연구에서 다양한 관점으로 정의되어왔다. Mayer(1987)는 지식, 과정 지식, 확산적 사고

과정, 비판적 사고과정을 창의적 문제해결을 구성하는 요소로 제시하였다. 또한 배영권(2006)은 창의적 문제해결력 신장을 위한 로봇 프로그래밍 교육 모형을 설계하는 연구에서 창의적 문제해결력이란 ‘문제를 해결하는 과정에서 지식, 기능, 비판적 사고, 확산적 사고, 동기 등이 복합적이고 역동적으로 상호작용 하여 해결방안이나 산출물을 만들어 내는 능력’으로 정의하였으며, 조연순 외(2008)는 창의적 문제해결이란 창의성이 발휘되는 과정 또는 여러 수행들 중 일부이며, 동시에 문제해결 중에서 창의성이 요구되는 특별한 형태로서, 창의적 문제해결은 인지적 사고기능(확산적 사고, 수렴적 사고)과 지식, 동기적 요소와 환경적 요소가 상호작용해 문제를 새롭게 발견하고 적절한 탐색과정을 거쳐서 새롭고 유용한 해결책을 생성해 가는 과정이라고 하였다.

창의적 문제해결에 대한 다양한 연구에서 도출된 정의와 특징을 종합해보면, 창의적 문제해결은 첫째, 비구조화된 문제를 해결하는 과정이고, 둘째, 문제를 해결하는 과정에서 확산적 사고와 논리·비판적 사고와 같은 고등 사고력을 필수적으로 요구하며, 셋째, 다양한 아이디어를 생성하고 문제에 대한 해결책을 찾아내는 체계적인 사고과정이고, 넷째, 인지적 사고과정 뿐 아니라, 지식, 동기, 환경적 요소가 밀접한 관계를 가지고 서로 상호 영향을 미치는 과정으로 정리할 수 있다.

이러한 창의적 문제해결의 과정과 단계를 체계적으로 다룬 창의적 문제해결 모형은 다양한 학자들에 의하여 해하고 제안되어 왔다(<표 II-1> 참고). Wallas(1926)는 창의적인 사고를 네 단계로 나누어 준비단계(preparation stage : 문제를 해결하기 위해 준비하며 문제를 다양한 각도에서 살펴보고 이해하는 단계), 부화단계(incubation stage : 아이디어의 산출기이며 잠시 문제에서 한 발 물러서는 단계), 조명단계(illumination stage : 갑작스럽게 새로운 아이디어나 해결방법이 떠오르는 단계), 검증단계(verification stage : 해결방안의 타당성을 검증하는 단계)로 구분하였다. Osborn(1963)은 창의적 문제해결 과정을 사실 발견(Fact Finding), 아이디어 발견(Idea Finding), 해결안 발견(Solution Finding)의 세 단계로 제안하였으며, 이후 Parnes(1967)가 이를 발전시켜

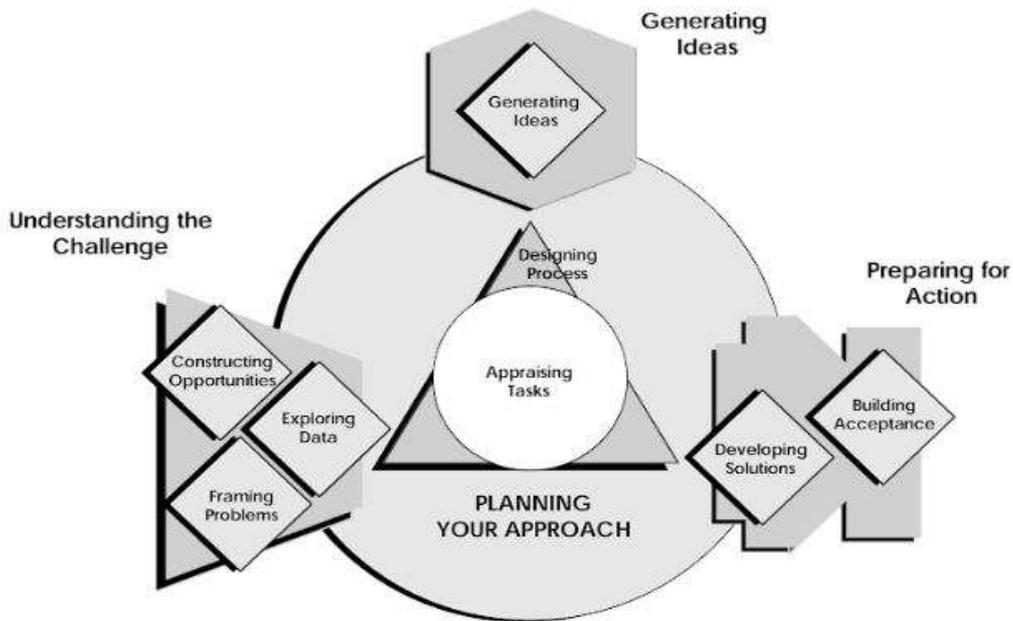
사실 발견(Fact Finding), 문제 발견(Problem Finding), 아이디어 발견(Idea Finding), 해결책 발견(Solution Finding), 수용(Acceptance Finding)의 다섯 단계로 제안하였다. 이후 Treffinger와 Isaksen의 모형(1985), Feldhusen와 Treffinger 모형(1993), Treffinger, Isaksen과 Dorval의 모형(2000), Puccio, Murdock과 Mance의 모형(2005)의 모형 등 많은 연구자들에 의해 계속적으로 연구되고 발전되어 왔다(Isaksen & Treffinger, 2004; Treffinger & Isaksen, 2005).

<표 II-1> 다양한 학자들이 제안하는 창의적 문제해결 모형과 단계

모형	창의적 문제해결 단계
Wallas (1926)	준비단계(Preparation stage)-부화단계(Incubation stage)-조명단계(Illumination stage)-검증단계(Verification stage)
Osborn (1963)	사실 발견(Fact Finding)-아이디어 발견(Idea Finding)-해결안 발견(Solution Finding)
Osborn-Parnes (Parns, 1967)	사실 발견(Fact Finding)-문제 발견(Problem Finding)-아이디어 발견(Idea Finding)-해결책 발견(Solution Finding)-수용(Acceptance Finding)
Treffinger & Isaksen(1985)	혼란 발견(Mess Finding)-자료 발견(Data Finding)-문제 확인(Problem Finding)-아이디어 발견(Idea Finding)-문제 발견(Solution Finding)-수용안 발견(Acceptance Finding)
Feldhusen (1993)	문제 산출-문제의 명료화-문제의 확정-아이디어 탐색-해결의 종합화-수행
Treffinger et al.(2000)	과제 파악(Understanding the Challenge)-과정 설계(Constructing Opportunities)-문제 확인(Framing Problems)-자료 탐색(Exploring Data)-문제 진술(Appraising Tasks)-아이디어 생성(Generating Ideas)-해결책 개발(Developing Solutions)-실행 조건 확립

	(Building Acceptance)
Puccio, Murdock & Mance(2005)	상황평가(Assessing the Situation)-비전탐색(Exploring the Vision)-도전 명료화(Formulating Challenges)-아이디어 탐색(Exploring Ideas)-해결안 명료화(Formulating Solutions)-수용안 탐색(Exploring Acceptance)-계획 명료화(Formulating a Plan)

창의적 사고 과정은 복잡하고 잘 정의되지 않은 문제를 관련된 개념의 조합과 재조직을 통해 새로운 아이디어로 해결해 나가는 과정이라고 할 수 있다(Scott, Leritz, & Mumford, 2004). 창의적 문제해결(Creative Problem Solving) 모형은 창의적 사고과정을 체계적으로 접근한 방법들 중 하나로, 창의성과 관련한 다양한 접근과 모형 중 가장 대표적인 것으로 평가되고 있다(Puccio, Firestein, Coyle, & Masucci, 2006; Treffinger et al., 2000).



[그림 II-3] CPS 6세대 모형(Treffinger et al., 2000)

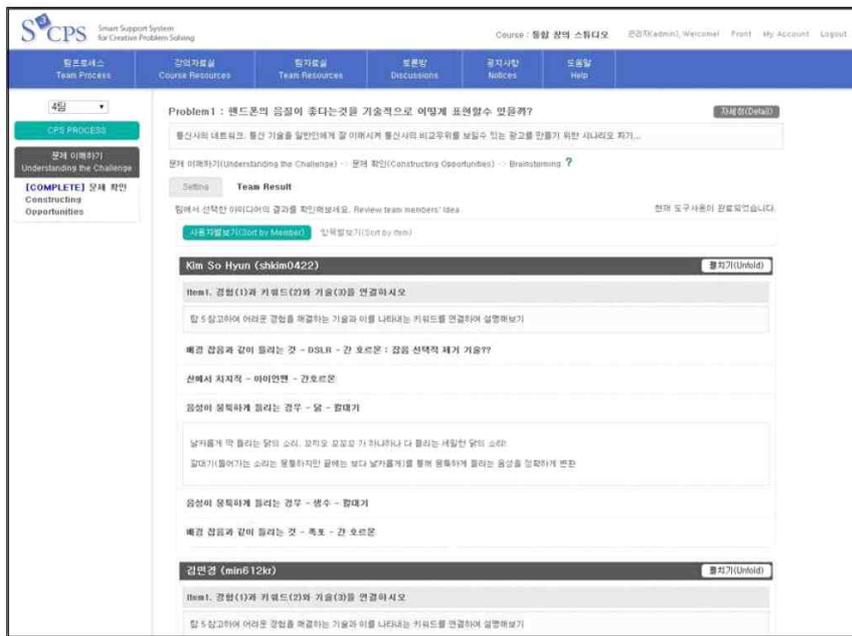
CPS모형은 1953년 Osborn이 CPS의 7가지 단계로 이루어진 모형을 제시한 이후, Osborn-Parnes(Parnes, 1967)의 5단계 모형,

Feldhusen(1993)의 모형, Treffinger, Isaksen과 Dorval의 모형(2000) 등 많은 연구자들에 의해 보완되고 발전되어 왔다. 최근 개발된 CPS의 버전 6.1 모형은 4개의 구성요소와 6개의 구체적인 단계로 이루어져 있으며, 4개의 구성요소는 크게 ‘관리요소’와 ‘과정요소’로 나누어진다(Treffinger, Isaksen, & Dorval, 2000). ‘관리요소’는 문제해결자가 ‘과정요소’와 그 단계들을 분석하고 선택하는 것을 안내하는 ‘접근의 계획’요소이며, ‘과정요소’는 ‘도전의 이해’, ‘아이디어 생성’, ‘행위를 위한 계획’의 세 개의 요소를 포함한다(홍미영, ;Treffinger, Isaksen, & Dorval, 2000)([그림 II-3] 참조).

‘관리요소’는 모형의 가운데에 위치하여 세 개의 ‘과정 요소’와 모두 연결되어 있는데, 이는 ‘관리 요소’인 ‘접근의 계획’요소가 CPS모형을 사용함에 있어서 안내자의 역할을 하는 것을 의미한다(김영채, 1999). ‘접근의 계획’에는 ‘과제의 평정’과 ‘과제의 설계’라는 두 가지 단계가 포함되어 있다. ‘과제의 평정’ 단계에서는 CPS 모형을 사용하기 전에 과제에 관련된 인물, 성취하고자 하는 결과, 과제와 관련된 상황이나 맥락 등을 다양한 측면에서 탐색하여 CPS모형을 사용 여부에 대한 적합성을 결정한다. ‘과제의 설계’단계는 CPS모형을 사용하기 위한 과정을 구체적으로 계획하고 수립하는 준비 단계이다. ‘접근의 계획’ 단계 외에 ‘과정 요소’에는 ‘도전의 이해’, ‘아이디어의 생성’, ‘행위를 위한 준비’의 3개 구성요소가 포함된다. ‘아이디어 생성’ 요소에는 ‘아이디어의 생성’ 단계가 포함되며, ‘행위를 위한 계획’ 요소에는 ‘해결책의 개발’ 및 ‘수용토대의 구축’ 단계가 포함된다. 그리고 이들 단계보다 더 구체적인 수준으로 CPS의 각 단계에는 발산적 사고와 수렴적 사고의 두 가지 국면이 있어 이들 간의 균형있는 사용을 강조한다(김영채, 2019; 황지현, 2010; Treffinger, Isaksen, & Dorval, 2000).

이상과 같은 CPS 모형은 다양한 아이디어를 생성하고, 독창적이고 합리적인 해결안을 도출하는 사고과정이며 발산적 사고와 수렴적 사고를 반복적으로 사용하는 것이 주요 특징이다(Treffinger, Isaksen, & Dorval, 2000). 특히 발산적 사고와 수렴적 사고의 전략을 지원하기 위해

S<sup>3</sup>CPS(Smart Support System for Creative Porblem Solving)와 같은 온라인 창의적 문제해결 지원 도구를 활용할 수 있다. S<sup>3</sup>CPS는 사용자가 문제를 창의적으로 해결하기 위한 활동 및 수행을 지원하는 온라인 시스템이다(임철일 외, 2014). S<sup>3</sup>CPS가 제공하는 사고 도구의 성격은 크게 발산적 사고와 수렴적 사고로 구성되어 있으며, 이러한 발산적 사고와 수렴적 사고를 돕기 위한 지원 도구로서 브레인스토밍, HIT, PMI, 평가행렬표 등의 사용이 가능하도록 지원하고 있다.



[그림 II-4] S<sup>3</sup>CPS에서의 학습자 활동 모습(임철일 외, 2014)

## 2. 컴퓨팅 사고력과 소프트웨어 교육

### 가. 컴퓨팅 사고력과 소프트웨어 교육

#### 1) 컴퓨팅 사고력의 개념과 구성요소

컴퓨팅 사고력(Computational Thinking)은 1980년대에 Seymour

Papert(1980)가 LOGO라는 프로그래밍 언어를 학습자에게 적용하여 학습의 또 다른 측면을 보여주는 연구과정에서 처음 사용되었으며, 이후 2000년대에 들어 Wing에 의해 본격적으로 알려지게 되었다. Wing은 컴퓨팅 사고력이란 문제를 식별하고, 문제를 제기하며, 문제 해결을 위해 확신을 갖고 꾸준히 노력하는 태도와 능력이라고 소개하며, 읽기, 쓰기, 셈하기와 마찬가지로 21세기를 살아가는 모든 사람들이 갖추어야 할 기본적인 사고 능력이라고 주장하였다(Krauss & Prottzman, 2017; Wing, 2006; Wing, 2008). 컴퓨팅 사고력을 21세기를 이해하기 위한 인간의 인지 능력의 확장이라고 본 Bundy(2007)와 사고력의 패러다임으로 접근한 Kramer(2007)의 연구에서도 컴퓨팅 사고력을 하나의 사고 능력으로서 강조하고 있다.

이후 2009년 NRC(National Research Council)의 주도로 진행된 워크숍에서 컴퓨팅 사고력에 대해 다양한 학자들의 접근이 이루어졌으며, 논의된 정의를 정리하면 다음 <표 II-2>과 같다(안상진, 서영민, 이영준, 2012; 류미영, 2015; NRC, 2010).

<표 II-2> 컴퓨팅 사고의 다양한 정의

학자	정의
Lee, P.	실제적인 적용을 위한 지능의 메커니즘에 대한 연구
Wulf, B.	절차에 초점을 맞추고, 절차적으로 해결하기 위해 현상을 추상화 하는 것
Abramson, D.	컴퓨터적인 기호 시스템을 사용하여 지식을 명료화하고, 암묵적 지식을 구체화하고, 그러한 지식을 컴퓨터적인 형태로 구현하고, 그런 과정을 통해 발생한 결과물을 관리 하는 것
Sussman, G.	어떠한 행동을 하는 데 있어 정확한 방법들을 조직하는 방식
Fox, E.	인간이 세상과 만나고, 과정을 사고하고, 디지털 표현을 다루는 일들

이후 컴퓨팅 사고력에 대한 정의를 학교 교육에 효과적으로 적용할 수 있는 기반을 마련하고자 ISTE와 CSTA에서는 K-12 교육을 위한 컴퓨팅 사고력의 조작적 정의를 다음과 같이 설명하였다(ISTE & CSTA, 2011).

- 1) 문제해결을 돕는 컴퓨터나 다른 도구를 사용할 수 있도록 문제를 만듦
- 2) 자료를 논리적으로 배치하고 분석하기
- 3) 모형과 모의실험과 같은 추상화를 통하여 자료를 표현하기
- 4) 절차적 사고력을 통하여 해결 과정을 자동화하기
- 5) 가장 효과적이고 효율적으로 목표를 달성하기 위한 가능한 해결책을 확인, 분석, 실행하기
- 6) 문제해결과정을 폭넓은 분야의 문제로 일반화하고 전이하기

컴퓨팅 사고력에 대한 논의는 학자에 따라 컴퓨팅 기술을 활용할 수 있는 역량에 초점을 두는 경우나 문제에 대한 절차적 사고력에 초점을 둔 문제해결 능력의 측면을 강조하는 경우 등 다양하게 이루어졌다. 특히 컴퓨터를 포함 한 ICT 도구 활용 능력 보다는 문제를 해결하는 사고 과정에 초점을 맞추어 지는 것으로 인식이 바뀌고 있음을 확인할 수 있다(김진숙 외, 2015). 류미영과 한선관(2018)은 정보교육 교사들을 대상으로 한 컴퓨팅 사고력에 대한 인식 연구에서 컴퓨팅 사고력의 정의에 관해 ‘컴퓨팅 사고력은 컴퓨터를 활용하여 문제를 해결하는 사람들처럼 사고하는 것’이라는 응답이 매우 높게 나오고, ‘컴퓨팅 사고력은 정보통신기술(ICT) 활용 능력의 일종이다’라는 항목에 대한 응답은 가장 낮게 나와 교사들의 인식이 이미 컴퓨팅 사고력은 정보통신기술의 기초 활용 능력보다는 코딩과 컴퓨터 과학의 지식 이해를 통해 문제를 해결하는 사고력 신장에 중점을 두고 있다는 분석 결과를 제시하였다.

Wing의 연구에 의하면 컴퓨팅 사고력의 핵심은 크게 추상화와 자동화의 두 가지 구성요소로 이루어져 있다. 추상화는 복잡한 자료 및 모듈,

시스템으로부터 핵심적인 개념 또는 기능을 간추려 내는 것을 의미하며, 자동화는 추상화 과정을 통해 만들어진 해결 모델을 컴퓨터가 이해할 수 있는 프로그래밍 언어로 표현하여 해결과정을 알고리즘화 하고 시뮬레이션을 실시하는 것을 말한다(김병조, 전용주, 김지현, 김태영, 2016; Wing, 2006; Wing, 2008)

Barr와 Stephenson(2011)은 컴퓨팅 사고력을 K-12 교육과정에 도입하기 위해 컴퓨팅 사고력의 구성요소를 11개로 제시하였으며(전용주, 2016), ISTE & CSTA(2011)는 Barr 외(2011)연구결과를 토대로 컴퓨팅 사고의 핵심 개념과 능력으로 자료 수집(data collection), 자료 분석(data analysis), 자료 표현(data representation), 문제분해(problem decomposition), 추상화(abstraction), 알고리즘 및 절차(algorithm & procedures), 자동화(automation), 시뮬레이션(simulation), 병렬화(parallelization)의 총 9가지 주요 개념으로 구분하였다. ISTE와 CSTA(2011)에서 제시한 컴퓨팅 사고력의 9가지 주요 개념과 정리하면 아래 <표 II-3>와 같다.

<표 II-3> 컴퓨팅 사고력의 주요 개념

개념	정의
자료 수집 (data collection)	문제의 이해와 분석을 토대로 문제를 해결하기 위해 자료 모으기
자료 분석 (data analysis)	수집된 자료와 문제에 주어진 자료를 세심히 분류하고 분석하기
자료 표현 (data representation)	문제의 내용을 적절한 그래프, 차트, 글, 그림 등으로 표현하기
문제 분해 (problem decomposition)	문제를 해결해나가기 위해 문제를 나누어 분석하기
추상화 (abstraction)	문제의 복잡도를 줄이기 위해 기본 주요 개념의 정의를 설정하기
알고리즘 및 절차 (algorithm & procedures)	지금까지의 문제를 해결하기 위한 과정을 순서적 단계로 표현하기

procedures)	
자동화 (automation)	순서적으로 나열하고 표현한 내용을 컴퓨터가 기기를 이용하여 해결과정의 최선책을 선택하기
시뮬레이션 (simulation)	복잡하고 어려운 해결책이나 현실적으로 실행이 불가능한 해결책을 선택하기 위한 모의 실험하기
병렬화 (parallization)	공동의 목표를 달성하기 위한 작업을 동시에 수행하기

한편 교육부에서도 2015년 개정 교육과정을 통해 컴퓨팅 사고력의 구성요소를 9단계로 구분하여 제시하였다. 컴퓨팅 사고력은 크게 추상화, 자동화 단계로 구성된다. 추상화는 실생활 문제를 컴퓨터를 통해 해결 가능한 형태로 표현하기 위한 사고 과정이다. 이를 위해 문제해결에 필요한 자료를 수집 및 분석하고 도표, 그래프 등을 활용하여 자료나 정보를 논리적으로 구조화하여 눈으로 보기 쉽게 나타낸다. 또한, 문제를 구성하고 있는 복잡한 요소를 작은 단위로 분해하고, 문제해결에 필요한 핵심 요소를 추출하여 적절한 해결책을 설계하는 모델링과정을 거친다. 그리고 문제해결 과정을 알고리즘으로 표현한다. 자동화는 추상화를 통해 도출된 알고리즘을 프로그래밍 언어를 이용하여 문제를 해결하는 코딩 과정을 거쳐, 프로그램을 실행하는 시뮬레이션 과정을 포함한다. 마지막으로 일반화를 통해 추상화와 자동화를 통한 문제해결 과정을 다른 문제에 적용시킨다(교육부, 2015a). 다양한 연구자와 기관에 의해 구분된 컴퓨팅 사고력의 구성요소는 <표 II-4>와 같다.

<표 II-4> 컴퓨팅 사고력의 구성요소 비교

Wing (2008)	Barr & Stephenson (2011)	ISTE & CSTA (2011)	교육부 (2015a)
추상화	자료수집(data collection)		
	자료분석(data analysis)		

	자료제시(data representation)		구조화
	문제분해(problem decompositoin)		분해
	추상화 (abstraction)		모델링
	제어구조(control structure)	알고리즘 및 절차 (algorithm and procedures)	추상화
	알고리즘 및 절차(algorithm and procedures)		
	분석과 모델 확인		
	자동화(automation)		코딩
	병렬화(parallelization)		자동화
자동화	실험 및 검증 (testing and verification)	시뮬레이션 (simulation)	시뮬레이션
	시뮬레이션 (simulation)		일반화

이와는 좀 다른 관점으로 MIT Creative Computing에서는 프로그래밍에 초점을 두고 컴퓨팅 사고력을 컴퓨팅 사고력 Concept, 컴퓨팅 사고력 Practice, 컴퓨팅 사고력 Perspective의 세 가지 영역으로 나누어 <표 II-5>과 같이 컴퓨팅 사고력의 구성요소를 제시하고 있다(Brennan & Resnick, 2012).

<표 II-5> MIT Creative Computing 에서의 컴퓨팅 사고력 구성요소(Brennan & Resnick, 2012)

구분	컴퓨팅 사고력 구성요소
컴퓨팅 사고력 Concept	시퀀스, 반복, 이벤트, 병렬처리, 조건, 연산자, 데이터
컴퓨팅 사고력 Practice	실험과 반복, 테스트와 디버깅, 재사용과 재구성, 추상화와 모듈화
컴퓨팅 사고력 Perspective	표현하기, 연결하기, 질문하기

또한 영국의 computing 교과에서 제시하는 컴퓨팅 사고력의 구성 요소를 Concepts와 Approaches 두 가지 영역으로 구분하여 [그림 II-5]와 같이 제안하였다(Computing at School/Barefoot Computing, 2016). Concepts 영역에서는 논리, 알고리즘, 분해, 패턴, 추상화, 평가의 6가지 구성요소가 포함되며, Approaches 영역은 키퍼링, 창조하기, 디버깅, 유지하기, 협력하기 등 5가지의 구성요소가 포함된다. 이상과 같이 컴퓨팅 사고력의 구성 요소는 다양한 연구기관 및 학자들마다 제각기 다른 관점을 기반으로 제시되고 있다.



[그림 II-5] Barefoot Computing 에서의 컴퓨팅 사고력 구성요소

## 2) 소프트웨어 교육의 목적 및 현황

2016년 세계경제포럼(WEF: World Economic Forum)에서 정보통신기술(ICT)을 기반으로 하는 새로운 산업시대를 일컫는 ‘제 4차 산업 혁명’이 언급되었다. 제 3차 산업 혁명이 컴퓨터와 인터넷을 기반으로 한 하드웨어 중심사회였다면, 제 4차 산업 혁명은 초연결과 초지능을 기반으로 하는 소프트웨어 중심사회이다(최소영, 2017; 황지은, 황성온, 2017).

이러한 패러다임 전환 속에서 새로운 융복합 역량을 갖춘 인재를 양성하기 위해 소프트웨어 교육의 중요성이 강조되고 있다. 2000년대 까지 정보 관련 교육은 ICT 활용교육과 관련된 교육이 주를 이루었다. 일반적인 정보화 기기의 사용법을 익히고 활용하여 교과에 접목시키는 방향으로 수업들이 이루어졌다. 이후 소프트웨어의 중요성이 증대됨에 따라 소프트웨어를 적극적으로 활용하여 문제를 해결하는 능력이 중요시되기 시작하였다. 특히 소프트웨어 교육이 기존의 ICT교육과 다른 점은 단순히 디바이스나 기기를 다루는 소양을 넘어 소프트웨어가 돌아가는 시스템과 원리를 익히고 이를 문제해결에 적극 활용하는 데 초점을 둔다는 점이다(홍정미, 2015)

소프트웨어 교육의 목적은 정보과학적 사고력을 길러 창의적으로 문제를 해결하는 능력을 배양하는 것이다. 즉, 컴퓨팅 사고력을 향상시킴으로써 다양한 산업 분야에서 일어나는 문제들을 창의적으로 해결할 수 있는 미래 인재를 키우는 것이다(교육부, 2015a; 안성진, 2014). 이러한 미래 인재상에 대해 교육부(2015a)는 [그림 II-6]과 같이 제시하고 있다.

## 컴퓨팅 사고력을 가진 창의·융합 인재



[그림 II-6] 소프트웨어 교육의 인재상(교육부, 2015a)

국내에서의 소프트웨어 교육은 2015년 개정 교육과정에 따라 2018년부터 초·중·고등학교에서 운영되고 있다. 교육부(2015a)에서 발간한 소프트웨어 교육 운영 지침에서는 학교급간 소프트웨어 교육의 연계성을 유지하고, 단순 ICT활용 교육을 넘어 소프트웨어로 새로운 사회의 가치를 창출하는 컴퓨팅 사고력을 갖춘 인재를 양성하는 것을 소프트웨어의 교육의 목표로 제시하였다(전용주, 2017). 소프트웨어 교육과정의 주요 개편 사항은 <표 II-6>과 같다.

<표 II-6> 2015 개정 교육과정 주요 개편 사항  
(교육부, 2015a; 김진숙 외, 2015)

구분	현행	개편안	주요 개편 방향
초등학교 (’19년~)	실과 내 ICT 단원 (12시간)	실과 내 소프트웨어교육 실시 (17시간)	- 문제해결과정, 알고리즘, 프로그래밍 체험 - 저작권 보호 등 정보윤리 의식 함양
중학교 (’18년~)	‘정보’과목 (68시간, 선택)	‘정보’과목 (34시간, 필수)	- 컴퓨팅 사고력 기반 문제 해결 교육

			- 알고리즘, 프로그래밍 개발
고등학교 (’18년~)	‘정보’과목 (심화선택 과목)	‘정보’과목 (일반선택 과목)	- 다양한 분야와 융합한 알고리즘 설계, 프로그램 개발 - 기초과목, 실무과목으로 개편하여 NCS 기반 교육과정 구성

한편 교육부(2015a)는 2015년 소프트웨어 운영지침을 통해 2009년 개정 교육과정 및 학교급간의 소프트웨어 교육의 연계성을 유지하고, 기존 초등학교의 ‘실과’ 과목 및 중학교의 ‘정보’과목을 학교급별로 새롭게 재구성한 방안을 제시하였다. 소프트웨어 교육 운영 지침상 학교급별 교육 목표는 <표 II-7> 과 같다.

<표 II-7> 학교 급별 소프트웨어 교육 목표(교육부, 2015a)

학교급 영역	초등학교	중학교	고등학교
생활과 소프트웨어	소프트웨어가 가져온 생활의 변화를 알고, 정보사회에 필요한 건전한 의식과 태도를 가진다.	소프트웨어 활용의 중요성을 알고, 정보 윤리의 개념을 이해하여 올바른 정보 생활을 실천하고, 정보를 교류할 수 있다.	컴퓨팅 기술과 융합된 다양한 분야를 이해하고, 정보 윤리를 실천하며, 정보기기를 올바르게 조작할 수 있다.
알고리즘과 프로그래밍	알고리즘과 프로그래밍을 체험하여 생활의 다양한 문제를 컴퓨팅 사고로 이해할 수 있다.	간단한 알고리즘을 설계하고 프로그램을 개발하여 문제를 해결할 수 있다.	알고리즘을 효율적으로 설계하고, 프로그램을 개발하여 창의적으로 문제를 해결할 수 있다.

컴퓨팅과 문제해결		컴퓨팅 사고력에 기반하여 실생활 문제를 해결할 수 있다.	컴퓨팅 사고를 기반으 로 다양한 분야와 융 합하여 문제를 해결 할 수 있다.

2015 개정 교육과정에서 제시한 소프트웨어 교육에서는 프로그램 개발 역량보다는 정보윤리의식과 태도를 바탕으로 실생활의 문제를 컴퓨팅 사고로 해결할 수 있도록 하는 것에 역점을 둔다. 이를 위해서 소프트웨어 교육의 핵심 역량으로 정보문화 소양, 컴퓨팅 사고력, 협력적 문제해결력의 세 가지를 제시하고 있다(김진숙 외, 2015). 정보문화 소양이란 정보사회의 가치에 대한 이해를 바탕으로 정보사회의 구성원으로서 윤리의식 및 시민의식을 갖추고 정보 기술을 활용하여 문제를 해결할 수 있는 능력을 말하며, 컴퓨팅 사고력은 실생활의 문제를 컴퓨팅 시스템을 활용하여 창의적으로 해결할 수 있는 능력을, 협력적 문제해결력은 학습자간 지식과 의사소통을 효과적으로 공유하고 협업을 통해 창의적으로 문제를 해결할 수 있는 능력을 의미한다.

특히 교수 프로그래밍 언어의 습득에 치중되지 않고 문제해결 절차에 초점을 두어야 하며, 컴퓨팅 사고는 정보 교육에 국한되는 것이 아니므로 다양한 교과, 특히 수학, 과학, 사회 교육 등에도 반영하여 지도해야 한다는 점을 교수·학습방법으로서 제시하였다는 점을 주목할 수 있다(교육부 2015a).

이와 같은 우리나라 소프트웨어 교육의 방향은 단순히 코딩 전문가를 양성하는 것이 아니라 주어진 문제를 분석, 파악하여 효율적인 해결 과정을 찾아보는 일련의 활동에서 컴퓨팅 사고력과 문제해결력을 향상할 수 있는 방향으로 실시되는 것으로 진행 되고 있다(홍정미, 2015). 따라서 창의적 문제해결력 증진을 위한 컴퓨팅 사고력 기반의 소프트웨어 교

육에 대한 연구가 필요하며, 나아가 디지털 사회의 필수적 요소인 컴퓨팅 사고력의 의미와 중요성을 학습자 스스로 인식하고 그 가치를 확인할 수 있도록 수업을 설계하고 원리를 탐구해 볼 필요성이 있다.

## 나. 창의적 문제해결을 위한 소프트웨어 교육의 수업 모형

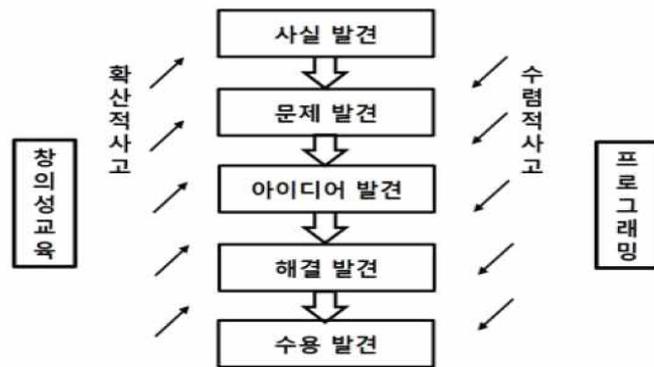
소프트웨어 교육 수업 모형은 수업의 목적, 교수자 및 학습자의 특성, 학습 내용, 교실 및 장비 등의 학습 환경 등의 다양한 요소를 고려하여 설계되어야 한다. 김진숙 외(2015)의 연구에서는 행동주의, 인지주의, 구성주의적 관점을 고려하여 학습 목표 영역인 지식, 기능, 태도 중 기능 영역을 중심으로 수업 모형을 개발하되, 기능 이외에 지식 및 태도 영역이 포함될 수 있도록 개발되었다(김진숙 외, 2015; 전용주, 2017). 특히 소프트웨어 교육 교수학습 모형은 컴퓨팅 사고력 신장이라는 목표를 달성하기 위해 각 단계별 활동 내에 컴퓨팅 사고력의 구성요소(분해, 패턴 인식, 추상화, 알고리즘, 프로그래밍)를 포함하도록 구성하는 등의 기본 방향을 설정하고 총 5개의 모형(시연중심 모형, 재구성 중심 모형, 개발 중심 모형, 디자인 중심 모형, CT요소 중심 모형)을 개발하여 제안하였다. 5가지로 특성화 한 수업 모형은 다음 <표 II-8>와 같다.

<표 II-8> 5가지 소프트웨어 교육 교수학습 모델  
(김진숙 외, 2015)

구분	절차	설명
시연중심모형 (DMM 모형)	시연(Demonstration)	교사의 설명과 시범, 표준모델 제시
	모방(Modeling)	학생 모방하기, 질문과 대답
	제작(Making)	단계적, 독립적 연습, 반복활동을 통한 기능 습득
재구성 중심모형	놀이(Use)	학습자 체험 활동, 관찰과 탐색

(UMC 모델)	수정(Modify)	교사가 의도적으로 모듈 및 알고리즘을 변형하여 제시
	재구성(reCreate)	놀이와 수정 활동을 확장하여 자신만의 프로그램 설계/제작
개발중심모델 (DDD 모델)	탐구(Discovery)	탐색과 발견을 통한 지식 구성
	설계(Design)	알고리즘의 계획 및 설계
	개발(Development)	프로그래밍 언어로 구현 및 피드백
디자인 중심모델 (NDIS 모델)	요구분석(Needs)	주어진 문제에 대한 고찰과 사용자 중심의 요구 분석
	디자인(Design)	분해와 패턴 찾기, 알고리즘의 설계
	구현(Implementation)	프로그래밍과 피지컬 컴퓨팅으로 산출물 구현
	공유(Share)	산출물 공유와 피드백을 통한 자기성찰
CT 요소 중심모델 (DPAA(P) 모델)	분해(Decomposition)	컴퓨터가 해결가능한 단위로 문제 분해
	패턴인식(Pattern Recognition)	반복되는 일정한 경향 및 규칙의 탐색
	추상화(Abstraction)	문제 단순화, 패턴인식으로 발견한 원리 공식화
	알고리즘(Algorithm)	추상화된 핵심원리를 절차적으로 구성
	프로그래밍 (Programming)	컴퓨터가 이해할 수 있는 언어로 구현/실행

한편 전성균 외(2010)의 연구에서는 프로그래밍 교육에서는 확산적 사고와 수렴적 사고를 경험하는 것이 가능하므로 창의적 문제 해결 모형과 프로그래밍 교육의 공통점을 찾을 수 있다고 하였다. 또한 기존의 프로그래밍 교육에서는 확산적 사고에 대한 비중이 적기 때문에, 확산적 사고를 중심으로 한 프로그래밍 교육을 통해 창의성을 강화해야 한다고 하였다. 특히 초등학생은 대부분 초급 프로그래밍 단계의 학생들이므로 효율성을 추구하는 것보다 경험 중심의 다양한 탐색 환경을 구현하는 것이 중요하다고 하였으며, 학습 과정을 다양하게 만드는 확산적 사고가 초등학생의 발달 단계에서 매우 중요한 요소라고 하였다. 따라서 학생들의 창의성을 신장시키기 위해서는 보다 적극적인 전략이 필요하며, 창의적 문제 해결 모형을 프로그래밍 수업 설계의 기본 모형으로 선택한 후, 확산적 사고를 촉진하기 위한 전략(브레인스토밍, SCAMPER, PMI)을 교수, 학습 과정에 포함시켜 학습 효과를 극대화하고자 하였다([그림 II-7]참고).



[그림 II-7] 창의적 문제 해결 모형, 창의성 교육, 프로그래밍 교육의 연관성(전성균 외, 2010).

전용주(2017)는 그의 연구에서 소프트웨어 및 컴퓨팅에 관련된 사고과정과 원리를 실생활의 소재와 능동적으로 관련지어 창의적으로 그 해결 방안을 구현해가는 과정으로 제시할 수 있는 수업 구성 원리로서 CT-CPS 수업 모형을 개발하였다. CT-CPS 수업 모형의 수업 과정은

문제를 스스로 찾는 것에서 출발하며, CPS에서 추구하는 문제발견/혼란 발견의 취지를 계승하여 실생활 또는 해당 교과에 관한 다양한 문제를 스스로 인식하고, 이를 문제화 하는 과정을 중시한다. 이 과정에서 CT의 학습요소인 자료수집, 자료분석, 자료제시 등을 활용하게 된다. CT-CPS 수업모형의 교수학습 단계와 활동은 다음 <표 II-9>와 같다.

<표 II-9> CT-CPS 수업모형(전용주, 2017)

교수학습단계	교수학습 활동
	주 활동 및 학습 매체
문제인식 및 분석 Problem Identification and Analysis	동기유발 및 수업에서 활용할 소프트웨어 도구 선택하기
	문제 인식하기(발견하기) - 문제화 하고 싶은 대상에 관한 자료 수집 (인터뷰, 설문조사, 답사, 조사 등) - 수집된 자료를 분석, 분류하여 시각화하기(* 표, 그래프, 그림 등)
	문제화 하기 - 이 문제를 해결하는 이유(목적), 문제에 관련된 지식, 문제의 가치 - 최종적으로 해결하고자 하는 문제 표현하기(* 글, PPT 등)
아이디어 구상 Idea Thinking	소프트웨어를 통해 문제를 해결할 수 있는 아이디어를 다양하게 생각하기 - 브레인스토밍, 마인드맵, 스케치 등
	해결 아이디어를 단순화하기, 핵심요소 찾기 - 해결 가능한 수준, 추상화
설계 Design	해결 아이디어의 설계 - 앱/프로그램 등을 장면별로 스케치해보기(* 종이, PPT 등 활용) - 논리적인 순서도 작성해보기
	시각과 논리를 종합한 전체 스토리보드 작성하기

구현 및 평가 Implementation and Evaluation	해결 아이디어의 구현 - 스토리보드를 기반으로 작성된 앱/프로그램의 초안 작성해보기(* 소프트웨어) - 디버깅하기(교사/동료와 함께)
	발표 및 시연하기(* 소프트웨어 및 PPT)
	평가/피드백 나누기 - 교사 관찰평가/동료평가/자기평가 등

한편 한선관(2015)은 창의컴퓨팅 교수학습모형을 통해 네 가지 단계를 제시하였다. 창의컴퓨팅은 문제를 해결하기 위한 디자인, 컴퓨팅, 융합의 과정을 거치며 창의적으로 문제를 해결할 수 있는 소프트웨어를 산출하는 활동이다. 창의컴퓨팅 교육의 목표는 창의적 문제해결력을 신장시키고 실생활의 문제를 해결하기 위한 디자인 능력을 기르며, 타 분야와의 융합을 통해 보다 넓은 시야를 갖게 하여 인류의 행복과 발전에 기여할 수 있는 소프트웨어를 산출하는 것이다. 창의컴퓨팅을 실제 학교에 적용하려면 교육내용은 있지만 교수학습 모형에 따른 교육전략이 필요하다. Havard Graduate School of Education(2014)은 창의컴퓨팅의 원리를 다음과 같이 4가지로 제시하여 각 단계에서 활용 가능하도록 제시하였다 ([그림 II-8] 참고).



[그림 II-8] 창의컴퓨팅의 원리(Havard Graduate School of Education, 2014)

창의 컴퓨팅은 초등학교 저학년, 초등학교 고학년, 중학생 이상에 적

합한 세 가지 모델을 제시하고 있다. 초등학교 저학년용 모델은 U-M-C(Use-Modify-Create) 모델로, 놀이 또는 활용을 체험한 후 수정, 확장 및 보완을 거쳐 최종적으로 배운 것을 토대로 새로운 산출물을 만드는 모델이다. 초등학교 고학년용 모델은 D-D-C(Discovery-Design-Create) 모델이며, 첫 번째로 문제에 도전하고, 디버깅 후 컴퓨팅 활동을 하는 Discovery 단계, 두 번째로 컴퓨팅 사고력 디자인과 Design Thinking을 수행하는 Design 단계, 마지막으로 배운 것을 토대로 새로운 산출물을 생산하고, 컴퓨팅과 융합하는 Create 단계로 구성된다. 중학생 이상 모델은 D-C-C(Design-Computing-Convergence) 모델로 문제를 디자인하고 언플러그드 컴퓨팅, EPL 컴퓨팅, 피지컬 컴퓨팅 등의 활동을 통해 최종적으로 융합하는 것이 특징이다.

<표 II-10> 창의 컴퓨팅 교수학습 모형

모델	대상	단계	활동
U-M-C 모델	초등학교 저학년	Use	- 놀이 또는 활용
		Modify	- 수정, 확장, 보완
		Create	- 배운 것을 토대로 새로운 산출물 만들기
D-D-C 모델	초등학교 고학년	Discovery	- 도전(챌린지) - 디버깅 - 컴퓨팅 활용
		Design	- Computational Thinking 디자인 - Design Thinking
		Create	- 배운 것을 토대로 새로운 산출물 생산 - 컴퓨팅과 융합
D-C-C 모델	중학생 이상	Design	- 디자인
		Computing	- 언플러그드 컴퓨팅 - EPL 컴퓨팅

			- 피지컬 컴퓨팅
		Convergence	- 융합

### 3. 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업 설계

#### 가. 융합의 정의와 융합인재교육의 특징

##### 1) 융합의 의미와 특징

개별 학문지식만으로 현대 사회의 다층적인 문제를 해결하기 어려워지면서 분야 간 경계 허물기와 지식의 융합을 통한 학문이 대세를 이루고 있다(임유정, 2012). 융합에 대한 개념과 정의는 학자마다 다양하다. 융합의 개념은 융합(convergence), 통합(integration), 통섭(consilience) 개념들과 구분 없이 사용되는 경우가 많다(노상우, 안동순, 2012; 목광수, 손정우, 배성문, 홍석영, 김용진, 2014; 이재경, 2017; 정정호, 2013)

‘융합’은 ‘다른 종류의 것이 녹아서 서로 구별이 없게 하나로 합하여지거나 그렇게 만들. 또는 그런 일’로 정의될 수 있다(“융합” in 국립국어원 표준국어대사전). 즉, 융합은 서로 다른 이질적인 여러 요소들이 일정한 원리나 절차에 따라 의도적으로 결합하여 새로운 하나의 통일체를 형성하는 것을 의미한다(이운발, 2005). 특히 융합의 개념은 여러 개의 사물을 물리적으로 또는 화학적으로 분해한 후에, 다시 복잡한 하나의 새로운 사물로 창조함으로써 기존의 기술이나 제품을 창조적으로 재조합하여 영역들 간의 시너지를 창출하는 작업이라 할 수 있다(신재한, 최은경, 2015).

‘통합’의 사전적 의미는 ‘부분을 연결시켜 전체를 만드는 일, 또는 이질적인 것을 하나의 체계 속에 융합시키는 일’로 정의된다(유승희, 성용

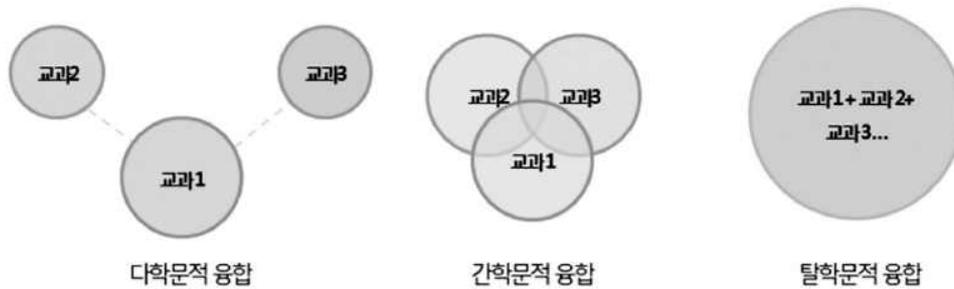
구, 2001). 통합은 교과간의 연계를 통해 특정 도구로 원하는 목표를 달성하고자 하며 학습의 효율성 및 탄력적인 운영에 도움을 주고 학습 내용을 종합적으로 적용하고 공통점을 추출하는데 목적이 있다(임유정, 2012).

마지막으로 ‘통섭’에 관하여 Wilson(2005)은 ‘융·복합과 같이 구체적인 합하여짐의 양상을 지적한 용어가 아니라, 거대 학문이라 할 수 있는 인문학과 자연과학, 사회과학 등이 상호보완적으로 융·복합 되어야 한다는 방향과 거시적 담론’이라고 표현하였다. 즉, 통섭은 원래의 학문이 사라지는 것이 아니라 학문들 간에 잦은 소통을 하는 것이다(김진수, 2012). 새로운 영역을 창조할 수 있는 융합 개념 속에 여러 개 흩어져 있던 분절적인 영역을 하나로 합친 단순한 통합의 개념도 포함할 수 있기 때문에, 융합의 가장 낮은 단계가 바로 통합이라 할 수 있다(이재경, 2017). 이와 같이 융합, 통합, 통섭 모두 서로 다른 것들을 한데 모은다는 의미를 갖고 있으나, 그 정도에 차이가 있음을 다른 용어로 설명하고자 한 것으로 이해할 수 있다(최소영, 2017). 이러한 융합의 의미와 특징을 살펴본 결과, 학문적으로 융합과 통합을 구분하여 논의하기도 하나 초·중등 교육과정에서 융합인재교육은 통합교육과 같은 맥락으로 적용되고 있다(김태형, 2015).

융합의 유형에 대한 다양한 학자들의 견해가 존재하지만 대표적으로 Drake와 Burns는 ‘다학문적 융합(Multidisciplinary Integration)’, ‘간학문적 융합(Interdisciplinary Integration)’, ‘탈학문적융합(Transdisciplinary Integration)’ 등의 접근으로 분류하였다(Drake & Burns, 2004).

다학문적 융합은 한 학문을 축으로 주위에 다른 학문을 배치하여 축에 있는 학문과 다른 학문들이 상호작용을 하게 하는 방식을 의미한다. 간학문적 융합은 어느 학문 또는 교과가 핵의 위치에 있는 것이 아니라 여러 학문의 개념, 방법, 절차가 문제의 해결에서 자유롭게 활용되는 방식을 의미한다. 그리고 탈학문적 융합은 간학문적 기능과 학문 분야별 기능을 실생활의 맥락에 적용하는 접근이며, 학문의 경계를 없애고 새로운 학문 영역으로 융합되는 방식을 의미한다([그림 II-9] 참조). 그들은 학

문이 연결되는 방식에 따라 교육 과정 통합의 형태를 분류했는데, 본질적인 차이는 ‘교과 영역 사이에 존재하는 분리의 정도’에 두고 있다(태진미, 2010). 교사는 교과를 융합할 때 수업의 목표 및 특성에 따라, 교사의 의도에 따라, 또는 현실적 여건에 따라 융합되는 정도를 조절해 볼 수 있다.



[그림 II-9] 학문간 통합 방식(김성원, 정영란, 우애자, 이현주, 2012)

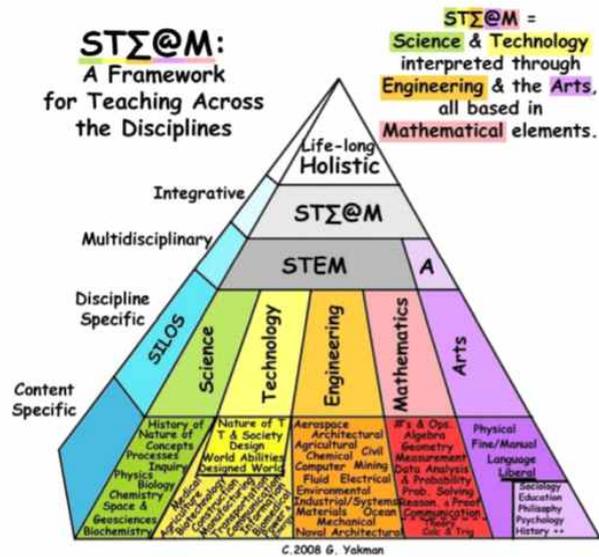
본 연구에서의 ‘융합’은 ‘다학문적 융합’에 해당하는 의미로 사용되었다. 즉, 한 학문을 축으로 두고 각 교과들과 관련 있는 주제나 이슈를 중심으로 교과를 연결하는 의미의 융합 교육이다. 따라서 교과-소프트웨어 융합 교육은 이러한 다학문적 융합을 통해 국어, 사회, 과학 등의 교과가 핵심적인 축을 담당하면서 컴퓨팅 사고력을 기반으로 한 소프트웨어 활용 활동을 통해 하나의 주제에 대해 문제를 해결해 나가는 교육을 의미한다.

## 2) 융합인재교육(STEAM)의 특징과 융합인재교육(STEAM)을 위한 교수설계

복잡한 사회 문제의 발생으로 다양한 교과지식을 기반으로 종합적이고 창의적인 문제해결력, 협력적으로 문제를 해결하는 능력, 그리고 현실을 비판적으로 재구성 할 수 있는 역량 등이 융합교육의 핵심 역량으로 강조되고 있다(백운수 외, 2011; Rennie, Venville, & Wallace, 2012). 교육

과학기술부(2011)에서는 창의적 인재를 육성하기 위한 정부의 정책으로 융합인재교육(STEAM)교육을 시행한다고 하였다. STEAM은 과학(Science), 기술(Technology), 공학(Engineering), 인문예술(Arts), 수학(Math)의 머릿글자를 조합한 용어로, 교과간의 통합적 접근방식을 의미하는 새로운 교육 틀이다(교육과학기술부, 2011). 융합인재교육(STEAM)은 과학, 기술, 공학, 수학, 예술 등의 교과지식과 방법들을 문제해결의 과정에서 통합적으로 학습하기를 기대한다(최소영, 2017)

이러한 다양한 교과에 대한 학습자의 흥미 향상을 위한 교육적 노력은 STEAM교육 이전에도 지속적으로 시도되었다. 여러 과목의 교과 지식을 융합적으로 가르치고자 하는 통합교육으로서 STS(Science, Technology, Social studies)교육, MST(Mathematics, Science, Technology)교육이 있었고, 21세기에는 STEM(Science, Technology, Engineering, Mathematics)교육이 시행되었다. STS교육은 Ziman(1980)이 처음 용어를 사용하고 1980년대 중반부터 시작된 미국의 과학통합교육의 흐름으로, 학교에서 가르치는 과학 내용이 실생활과 관련이 매우 적다는 비판이 제기됨에 따라 과학적 지식 외에 기술의 발달과 사회적 문제를 통합적으로 가르쳐보고자 하는 교육이다(김진수, 2012). MST교육은 수학, 과학, 기술의 세 교과를 통합한 교육이며, 주로 기술교과 중심의 통합으로 볼 수 있다(김영채, 2019; 김진수, 2012). 1997년 미국 뉴욕 주 교육부에서는 수학의 분석, 과학의 탐구, 공학의 설계를 MST 통합 교육의 표준으로 설정하고 실생활 문제해결 과정에서 수학, 과학, 기술 교과의 내용을 학습할 것을 제시하였다(강갑원, 2015). STEM교육은 MST교육에 공학(Engineering)을 포함한 교육이다. Sanders(2009)는 STEM교육을 기술과 공학의 설계를 중심으로 한 교육으로 정의하며, 기술과 공학 중 적어도 한 가지 분야를 포함하여 이루어지는 통합 교육을 통합적 STEM교육(integrative STEM)이라고 하였다. 즉, STEM교육은 기술과 공학 교육을 중심으로 한 통합적 노력이라고 볼 수 있다.



[그림 II-10] STEAM 피라미드(Yakman, 2008)

한편 미국의 Yakman(2008)은 융합인재교육(STEAM)을 구체적으로 정의하기 위해 STEAM 모형을 [그림 II-10]과 같이 제시하였다. Yakman은 STEM에 Arts를 추가 하여 Science, Technology, Engineering, Arts, Mathematics의 다섯 가지 전통적 학문 영역들의 통합적 구조화를 위한 교육학적 모델을 고안하고 이를 융합인재교육(STEAM)이라 명명하였다.

융합인재교육(STEAM)은 STEM에 Art에 해당하는 예술영역이 추가 되어 교육하는 형태로, 국내에서는 2011년 교육과학기술부가 ‘융합인재교육’으로 명명하였다. 융합인재교육(STEAM)의 목적은 학문 간의 차이로 인하여 합치기 어려웠으나 이들이 갖는 공통점을 바탕으로 해당 분야의 지식을 활용하여 문제 해결력을 기르는 차원에서 영역을 확장시키려는데에 있다(한국과학창의재단, 2012). 융합인재교육(STEAM)에 대해 2015 개정 교육과정 총론에서는 학교 교육을 통해 기르고자 하는 핵심역량에 ‘폭넓은 기초지식을 바탕으로 다양한 전문 분야의 지식, 기술, 경험을 융합적으로 활용하여 새로운 것을 창출하는 창의적 사고 역량’(교육부, 2015b)이라 하였으며, 더욱더 적극적으로 융합 교육을 할 것을 명시하였다. 한편, 창의성과 관련하여 추가된 Art의 학문분야는 일반적으로 예술

분야에 한정하여 논의되었으나, 인문학을 포함한 Arts로 확대 해석하여 학교에서 융합을 시도해야 한다는 주장이 제기되고 있다(노상우, 안동순, 2012; 목광수 외 2014). 인문 교과도 포함됨으로써 STEAM 교육은 전교과와 관련될 수 있다(최소영, 2017)

융합인재교육(STEAM)은 다양한 분야의 융합적 내용을 창의적 설계(Creative Design)와 감성적 체험(Emotional Touch)으로 경험함으로써 과학기술과 관련된 다양한 분야의 융합적 지식, 과정, 본성에 대한 흥미와 이해를 높여 창의적이고 종합적으로 문제를 해결할 수 있는 융합적 소양(STEAM Literacy)을 갖춘 인재를 양성하는 것이다(백윤수 외, 2012).

이러한 융합인재교육(STEAM)의 교육 프로그램 개발을 위해 한국과학창의재단(2012)에서는 ‘상황제시, 창의적 설계, 감성적 체험’을 학습 준거로 제시하였다. 상황 제시는 실생활의 문제를 자신의 문제로 인식할 수 있도록 동기를 부여하는 장치로 자연스러운 융합을 위해 활용될 수 있다. 창의적 설계는 실생활의 문제를 정의하고 최선의 해결책을 만들어가는 과정으로, 문제 정의 능력과 문제 해결 능력 증진을 목적으로 한다. 감성적 체험은 학습의 흥미와 동기 부여를 위한 요소로, 체험활동, 성공의 경험, 새로운 도전요소 및 자기 평가 활동을 포함한다(백윤수 외, 2012; 조향숙, 김훈, 허준영, 2012).

또한 백윤수 외(2012)는 STEAM교육의 특징으로 내용 융·통합, 창의적 설계, 감성적 체험의 세 가지를 강조하고, 이를 통해 ‘융합(Convergence), 창의성(Creativity), 의사소통(Communication), 배려(Caring)’의 역량을 기르는 4C-STEAM을 제안하였다. STEAM교육의 특징 중 내용 융·통합은 교과 간 통합에 대한 것으로, STEAM교육에서의 차별화된 특징이기 보다 통합교육 방법에 근거하고 있으며(김진수, 2012; 최소영, 2017), 이에 대해 Drake와 Burns(2006)는 다학문적, 간학문적, 탈학문적 융합을 제시하였다. STEAM교육을 통해 이룰 수 있는 핵심 역량인 4C 중 하나인 ‘융합’은 STEAM 융합 지식 이해 및 STEAM 융합지식 설계 및 활용, 응용 능력, STEAM 외의 맥락적 지식

의 이해 등을 포함한다. ‘창의성’은 창의력, 문제해결력, 문제확인능력, 정보수집능력, 정보분석능력, 의사결정능력, 평가능력 등의 요소를 포함하며, ‘의사소통’은 언어적 소통, 시청각적 소통, 학문적 능력, 소통 태도, 협력 등의 요소가 포함된다. ‘배려’는 자기애, 자신감, 자아정체감, 자기 효능감, 타인에 대한 배려 등과 같은 사회적 감성학습 요소를 포함한다 (김진수, 2012; 백윤수 등, 2012). 백윤수 외(2012)는 감성적 체험, 창의적 설계, 내용 융·통합의 STEAM 요소를 바탕으로 상황제시로 시작하여 창의적 설계와 감성적 체험을 포함하는 내용 융·통합 활동을 통해 보상의 경험이 새로운 문제에 도전하는 순환의 과정을 다음과 같이 제시하였다 ([그림 II-11] 참고).

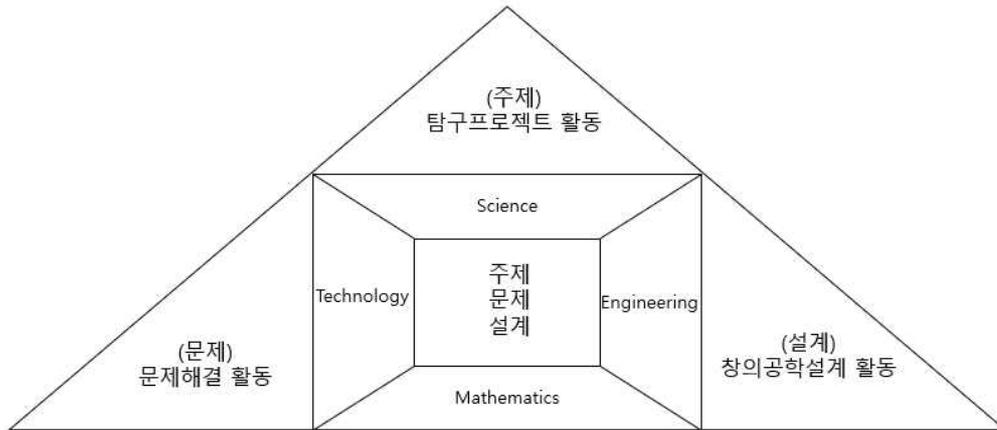


[그림 II-11] STEAM 교육 수업구성 원리(백윤수 외, 2012, p.68)

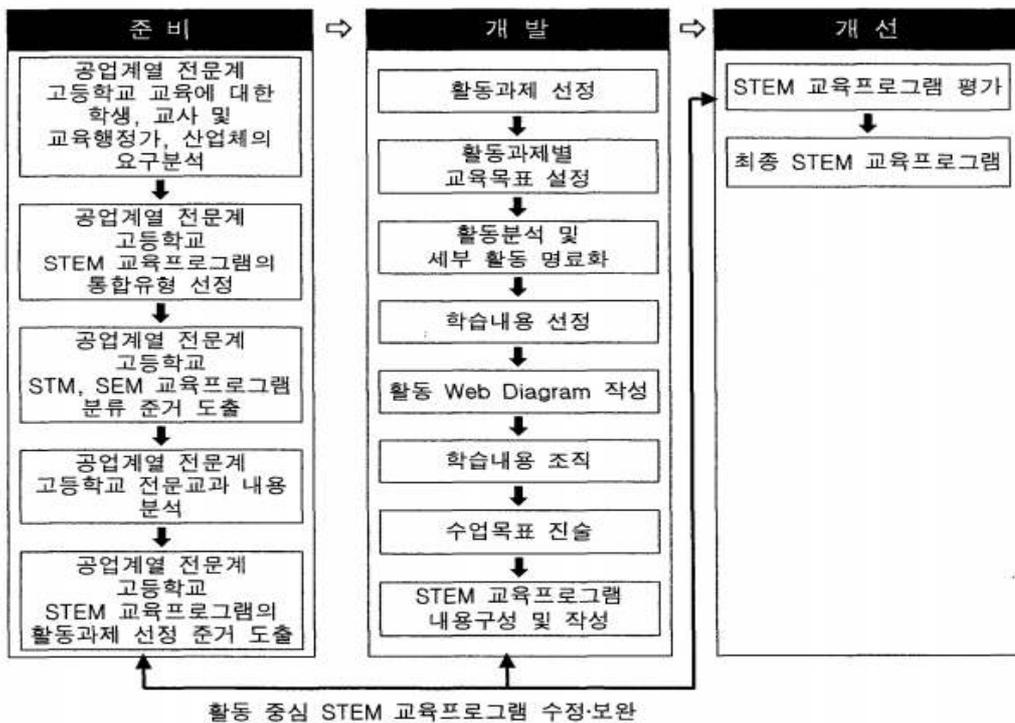
한편 우리나라는 4차 교육과정부터 초·중등학교에서 통합교육이 강조되면서 교과 간 통합에 기초한 STEAM교육을 위한 교수설계 및 교수설계 모형 개발 연구가 지속적으로 이루어지고 있다(김정호, 2015; 김진수, 2012; 이영태, 남창우, 2015; 문대영, 2008; 윤옥한, 2017; 이소이, 노태천, 2011; Stohlmann, Moore, & Roehrig, 2012). 교과-소프트웨어융합 수업설계를 위한 시사점을 얻기 위해 STEM, STEAM 교육의 교수설계 및 교수설계 모형에 대한 연구를 분석하였다.

문대영(2008)은 초중등학생 대상의 공학교육 프로그램 개발을 위해 활

용할 수 있는 개념 모형을 개발하였으며, 교과 간 통합을 위한 핵심 요소(주제, 문제, 설계)와 이에 대응되는 활동(탐구프로젝트활동, 문제해결 활동, 창의공학설계활동)을 제시하였다([그림 II-12] 참고).



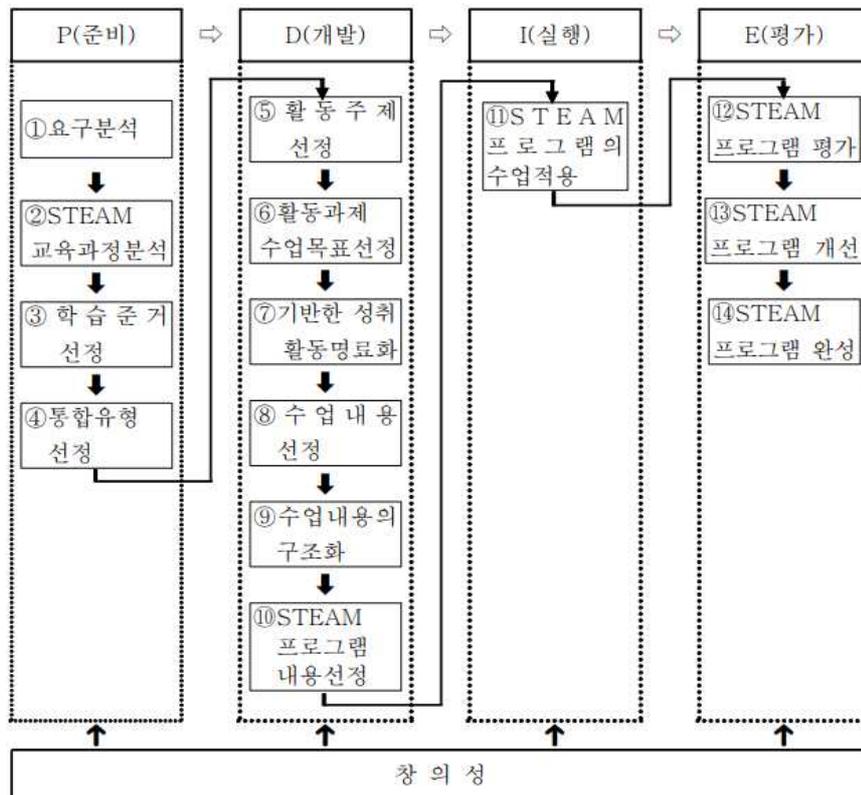
[그림 II-12] STEM 통합 접근의 교육 프로그램 개념 모형(문대영, 2008)



[그림 II-13] 활동중심 STEM 교육 프로그램 개발을 위한 절차모형(배선아, 2011)

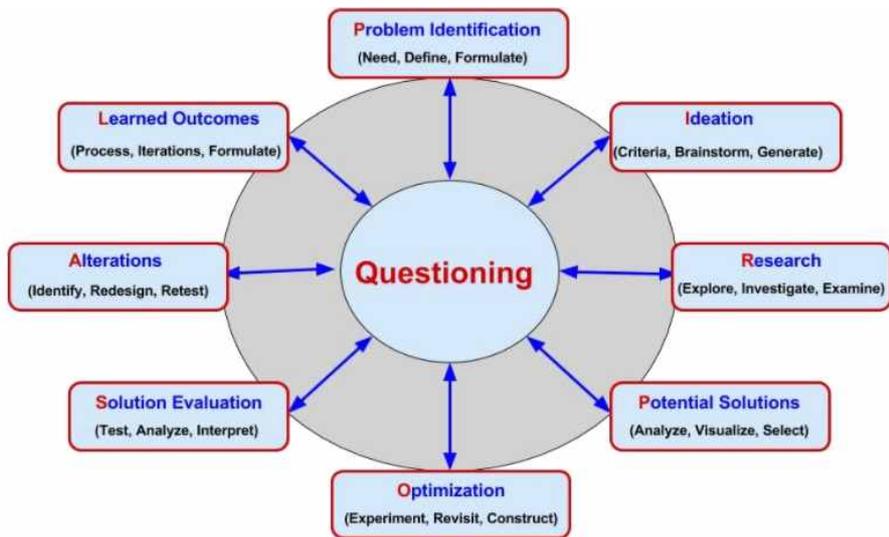
배선아(2011)는 기술, 공업 교육에서의 활동 중심 STEM 프로그램 설계 모형을 개발하였다([그림 II-13] 참고). 이 모형은 기본적으로 준비단계, 개발단계, 개선단계로 이루어져 있으며, 15단계의 세부 하위단계로 구성된다. 특히 수업 목표 설정이 가운데에 위치하여 학습내용을 우선적으로 선정하고 그 내용과 관련된 목표가 정해지는 것이 특징이다.

또한 STEAM 교육 프로그램을 체계적으로 개발하기 위해 김진수(2011)는 초중등학교의 STEAM 교육을 위한 수업 자료 개발을 위한 PDIE 절차 모형을 개발하였다. 이는 준비(Preparation), 개발(Development), 실행(Implementation), 평가(Evaluation)의 4단계 모형이다. PDIE 모형은 ADDIE 모형의 단점을 줄이고 장점만을 취한 모형으로, 특히 각 단계별로 학생들의 창의성 교육을 강조하였다([그림 II-14 참고]).



[그림 II-14] STEAM교육 프로그램 개발을 위한 PDIE모형(김진수, 2012)

한편 Wells(2015)는 통합적 STEM 교육을 위한 PIRPOSAL 모형을 개발하였다([그림 II-15] 참고). PIRPOSAL 모형은 문제 확인(Problem identification), 아이디어 생성(Ideation), 연구(Research), 가능한 해 찾기(Potential solutions), 최적화(Optimization), 해결안 평가(Solution evaluation), 대안(Alterations), 학습결과(Learning outcomes)의 8개 요소로 이루어져 있다. 특히 원 중심의 질문하기(Questioning)은 각 단계마다 학습자의 발산적 사고와 수렴적 사고를 돕는 역할을 하는 것이 특징이다.



[그림 II-15] PIRPOSAL 모형(Wells, 2015)

이상의 선행연구들을 종합해 보았을 때, 융합인재교육(STEAM)을 위한 교수설계는 교과를 융합하고, 실제생활의 문제를 제시하고, 다양한 설계 학습 활동을 강조하고 있다는 것을 알 수 있다. 특히 융합인재교육(STEAM)의 수업 활동을 위해 학습자에게 주어지는 융합 문제는 활동 목적, 학습자 수준, 흥미와 학습 환경을 고려한 최적의 문제를 개발할 필요가 있으며, 해결안이 다양하게 존재하는 비구조화된 문제를 개발할 필요성이 있다(최소영, 2017).

즉 융합인재교육(STEAM)의 교수설계는 핵심적인 교과의 아이디어를 중심으로 두 개 이상의 교과 내용을 유기적으로 통합하고, 학생들이 실

생활에서 경험할 수 있는 다양하고 복잡한 비구조 문제를 제시하며, 여러 가지 유형의 결과 및 산출물을 낼 수 있는 창의적 설계활동을 중심으로 구성되어 있음을 확인할 수 있다.

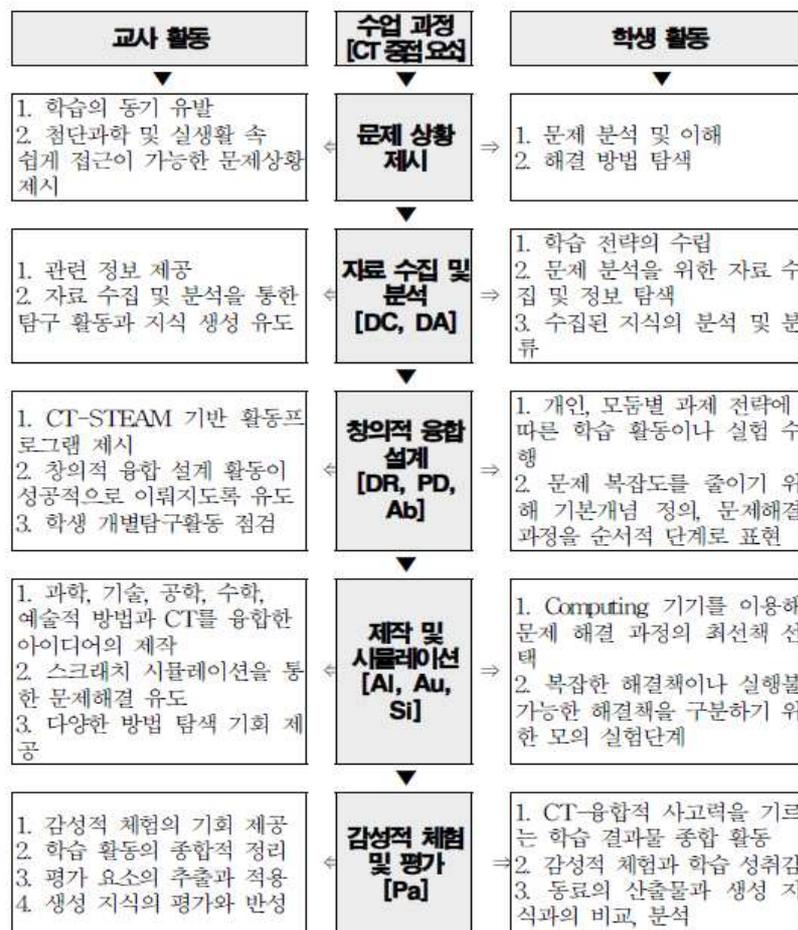
## 나. 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어 융합교육

일상생활의 복합적 문제를 해결하기 위한 창의적 사고력(박만준, 2010; 주미경, 문종은, 송륜진, 2012; Wilson, 2005)과 문제해결능력을 신장시키기 위한 방안으로 소프트웨어 교육과 개별 교과목을 접목한 교과-소프트웨어융합교육의 중요성이 대두되고 있다(박금주, 최영준, 2018; 이미화, 함성봉, 2018; 전성균, 이영준, 2014; Green & Anid, 2013). 교육부(2015a)에서도 다른 교과 지식과 소프트웨어를 융합하여 다양한 문제 상황을 해결하는 교육이 이루어질 필요가 있음을 강조하였다.

그동안의 교과-소프트웨어융합 교육은 주로 소프트웨어와 과학 교과 영역의 융합을 중심으로 이루어지고 있다(노희진, 백성혜, 2015; Hutchins et al., 2020; Lin, Wang & Wu, 2019; Sengupta, Kinnebrew, Basu, Biswas & Clark, 2013). 정예진(2018)은 초등 수학 영재 대상 컴퓨팅 사고력을 활용한 수학중심 STEAM 프로그램을 개발하여 적용한 결과 학습자의 융합적 사고력과 협업 역량을 신장시키는 데 효과적임을 밝혔다. 류미영(2015)은 컴퓨팅 사고력과 수학교과의 융합교육 프로그램을 문제해결 알고리즘 사고 모델, 컴퓨팅 시스템의 활용 능력 모델, 컴퓨터 과학의 지식 적용 모델을 준거로 프로그램을 개발하여 전문가 및 일반교사를 대상으로 효과성 검증을 진행한 결과, 프로그램의 효과성, 적용가능성, 문제해결능력, 추상화 능력, 자동화 능력, 교육목표 달성에 있어서 긍정적인 결과를 얻었다. 또한 Lin 외(2019)는 컴퓨터 교과목 수업(Technology)과 물리 교과(Science) 내용을 접목하여 STEM수업을 실시하였다. 그 결과 STEM 수업을 실시한 집단의 프로그래밍과 물리학 학습 성취도가 기존의 전통적 수업을 실시한 집단보다 높았으며, 더 높은

자기 효능감을 보인 것을 확인하였다. 한편 일부 연구에서 소프트웨어와 국어, 영어, 미술 등 과학 이외의 다양한 과목을 융합한 교육(박판우, 2016; 이민희, 임해미, 2013)도 보고되었다. 예컨대, 문희식(2014)은 국어, 사회, 수학, 과학, 실과 교과와 단원을 융합하여 융합인재교육(STEAM) 프로그램을 개발하였고, 이에 대해 학습자 만족도 및 학습 효과 측면에서 긍정적인 결과를 확인하였다.

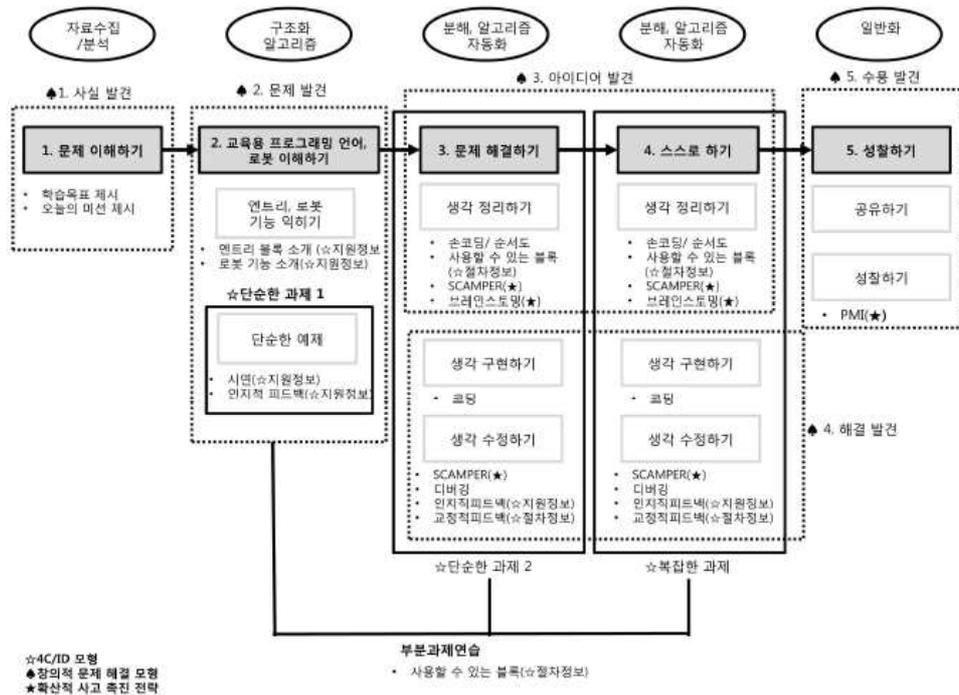
한편 함성진, 김순화, 박세영, 송기상(2014)은 초등학생을 대상으로 블록형 코딩 소프트웨어인 스크래치를 사용한 컴퓨팅 사고력 기반 융합인재교육(STEAM) 수업모형인 CT-STEAM 수업 모형을 개발하고 CT-STEAM 모형을 기반으로 교육 프로그램을 제작하였다([그림 II-16] 참고).



[그림 II-16] CT-STEAM 수업모형(함성진 외, 2014)

CT-STEAM 수업 모형은 컴퓨팅 사고력을 기반으로 한 미래형 융합 인재 교육을 위해 개발되었으며, ‘문제상황 제시’, ‘자료 수집 및 분석’, ‘창의적 설계’, ‘제작 및 시뮬레이션’, ‘감성적 체험’의 5단계를 가진다. 이 후 김순화, 함성진, 송기상(2015)은 CT-STEAM 수업 모형을 기반으로 제작한 교육 프로그램을 학습 현장에 적용시켜 학습자의 융합적 사고력과 컴퓨팅 사고력을 신장시키는 데 효과가 있음을 밝혔다.

이와 같은 연구를 발전시켜 컴퓨팅 사고력 기반 소프트웨어 교육을 통해 창의성과 창의적 문제해결력 신장을 목표로 학습 콘텐츠 및 교육과정을 개발하고 효과성을 검증하는 연구가 시도되고 있다. 김병조 외(2016)는 초등 정보융합영재를 위한 컴퓨팅 사고력 기반의 실생활 문제해결 수업 콘텐츠를 개발하고 이를 적용한 결과 창의적 성향의 하위 요소인 정보과학에 대한 흥미와 산출물의 창의성에 유의미한 차이를 보였고, 참신성, 독창성, 논리성 등의 하위 항목에도 유의미한 차이를 보임을 밝혔다. 노지예와 이정민(2018)은 4C/ID 모형과 창의적 문제해결 모형을 기반으로 컴퓨팅 사고력의 구성요소를 고려하여 로봇 활용 소프트웨어 교육 프



[그림 II-17] 로봇 활용 소프트웨어교육 프로그램의 구조

로그램([그림 II-17] 참고)을 설계하였으며, 프로그램 적용 결과 학습자의 컴퓨팅 사고력과 창의적 문제해결력이 유의하게 향상되었음을 확인하였다.

또한 권정인(2013)은 초등학생을 대상으로 컴퓨팅 사고력 기반의 창의적 문제해결 모델을 적용한 프로그램을 실시하였고, 컴퓨팅 사고력 기반의 CPS 모델의 교수학습이 창의적 문제해결의 정의적 요소 중 자기 확신 및 독립성, 비판적·논리적 사고, 동기적 요소에 유의미한 영향을 미치고 있음을 밝혔다.

지금까지 살펴본 창의성과 창의적 문제해결, 컴퓨팅 사고와 소프트웨어교육 그리고 융합인재교육(STEAM)에 대한 개념과 특징, 그리고 교과-소프트웨어융합 수업설계에 대한 선행 연구를 종합하여 볼 때, 컴퓨팅 사고력은 컴퓨터 과학에 한정된 개념이 아닌 모든 학문 분야와 실생활의 문제를 창의적으로 해결하기 위한 문제해결방법으로, 교과간의 학문적 융합으로 실생활의 복합적인 문제를 해결하는 것이 목표인 융합인재교육(STEAM) 및 교과 융합 교육과 합치하는 부분을 가지고 있으며(함성진 외, 2014), 창의·융합 인재를 육성하는 공통의 목표로 두고 있음을 알 수 있다(교육부, 2015a).

### Ⅲ. 연구 방법

본 연구는 창의적 문제해결을 위한 컴퓨팅 사고력 기반 소프트웨어 수업설계 모형 및 원리를 개발하기 위하여 설계·개발 연구(Design and Development Research) 방법론을 적용하였다. 설계·개발 연구 방법은 교수적 또는 비교수적 산출물 및 도구, 그리고 그 개발을 이끄는 새로운 모형의 생성에 관한 실증적 기반 확립을 목표로 하는 설계, 개발, 평가과정에 대한 체계적 연구로, 새로운 지식을 생성할 뿐만 아니라 기존 실천들을 타당화하고 교수설계의 이론기반을 실증적으로 확대하고 발전시키는 데 그 목적을 두고 있다(임철일, 2012, Richey & Klein, 2014).

Richey와 Klein(2007)에 의하면 설계·개발 연구는 과정이나 절차를 연구대상에 포함하며, 이론과 실제 사이의 교량 역할을 하는 인공물(artifact)을 연구결과로서 개발하는 목적을 가지고 있다(Ellis & Levy, 2010). 이 때 인공물이란 설계모형 및 설계원리, 설계전략, 설계지침 등을 포함한다. 따라서 이 설계·개발 연구는 설계, 개발, 평가과정에 대한 체계적인 연구로, 실제로 교육공학 전문가들이 학습을 촉진시키고 수행의 증진을 도모하는 데 필요한 비교수적 혹은 교수적 개입을 창출하고 관리하며 활용하는 다양한 제안들과 상응할 수 있다(Reiser, 2012; Richey, Klein, & Tracey, 2010).

설계·개발 연구는 산출물 및 도구 연구(products and tools research)와 모형 연구(model research)의 두 가지 범주로 구분된다. 산출물 및 도구 연구는 특정 산출물 및 도구 설계와 개발 프로젝트에 관한 연구로 맥락 의존적인 특징을 지닌 반면, 모형 연구는 새로운 설계와 개발 과정의 일반적인 분석을 지향하며, 산출물 연구보다는 좀 더 일반화 될 수 있는 특징을 가진다. 이 중 모형연구는 설계모형 뿐만 아니라, 설계원리, 설계전략, 설계지침 등을 개발하는데도 활용된다(Richey & Klein, 2007).

본 연구에서는 설계·개발 연구의 두 가지 유형 중에서 유형 2의 모형 연구를 사용하였다. 모형 연구의 탐색 주제는 첫째, 모형 개발연구

(model development research), 둘째, 모형 타당화연구(model validation research) 셋째, 모형 사용연구(Model use research)의 세 가지로 나뉠 수 있다(<표 III-1> 참고).

<표 III-1> 설계·개발 연구의 대표적인 두 가지 유형

설계·개발 연구	
산출물 및 도구 연구(유형 1)	모형 연구(유형 2)
<ul style="list-style-type: none"> <li>• 포괄적인 설계와 개발 프로젝트</li> <li>- 교수적 산출물과 프로그램</li> <li>- 비교수적 산출물과 프로그램</li> </ul>	<ul style="list-style-type: none"> <li>• 모형개발</li> <li>- 포괄적 모형개발</li> <li>- 모형 구성요소 과정 개발</li> </ul>
<ul style="list-style-type: none"> <li>• 구체적인 프로젝트 단계</li> <li>- 분석</li> <li>- 설계</li> <li>- 개발</li> <li>- 평가</li> </ul>	<ul style="list-style-type: none"> <li>• 모형 타당화</li> <li>- 모형 구성요소의 내적 타당화</li> <li>- 모형의 영향에 대한 외적 타당화</li> </ul>
<ul style="list-style-type: none"> <li>• 도구의 설계와 개발</li> <li>- 도구개발</li> <li>- 도구사용</li> </ul>	<ul style="list-style-type: none"> <li>• 모형 사용</li> <li>- 모형 사용에 영향을 주는 조건에 관한 연구</li> <li>- 설계자 의사결정연구</li> <li>- 설계자 전문성 &amp; 설계자 특성 연구</li> </ul>

\* 출처 : Richey & Klein(2007), p. 8에서 발췌함

모형 개발연구는 포괄적 모형을 개발한 뒤에 모형 구성요소 과정을 개발하는 연구이고, 모형 타당화 연구는 모형 구성요소의 내적타당화를 수행하고, 이후 모형의 영향에 대한 외적 타당화를 진행하는 연구이다. 모형 사용 연구는 모형 사용에 영향을 주는 조건에 관한 연구로, 설계자 의사결정에 대한 연구, 설계자의 전문성 및 설계자 특성에 관한 연구를 포함한다. 결국 설계·개발 연구의 유형 2 모형 연구는 모형과 과정 자체

에 초점을 두고, 기존 혹은 새롭게 만들어진 개발 모형, 과정, 기법의 타당성 또는 효과성 탐색 등을 목적으로 한다. 더불어 유형 2 모형 연구 유형은 성공적인 설계와 개발을 촉진시키는 조건을 확인하고 기술한다. 설계 모형이나 설계 원리를 개발하는 연구일수록 추상성이 증가하며, 설계 지침을 개발하는 연구일수록 구체성이 증가하는 특징을 가진다 (Reigeluth, 1983; Reigeluth & Carr-Chellmann, 2009; Richey & Klein, 2007).

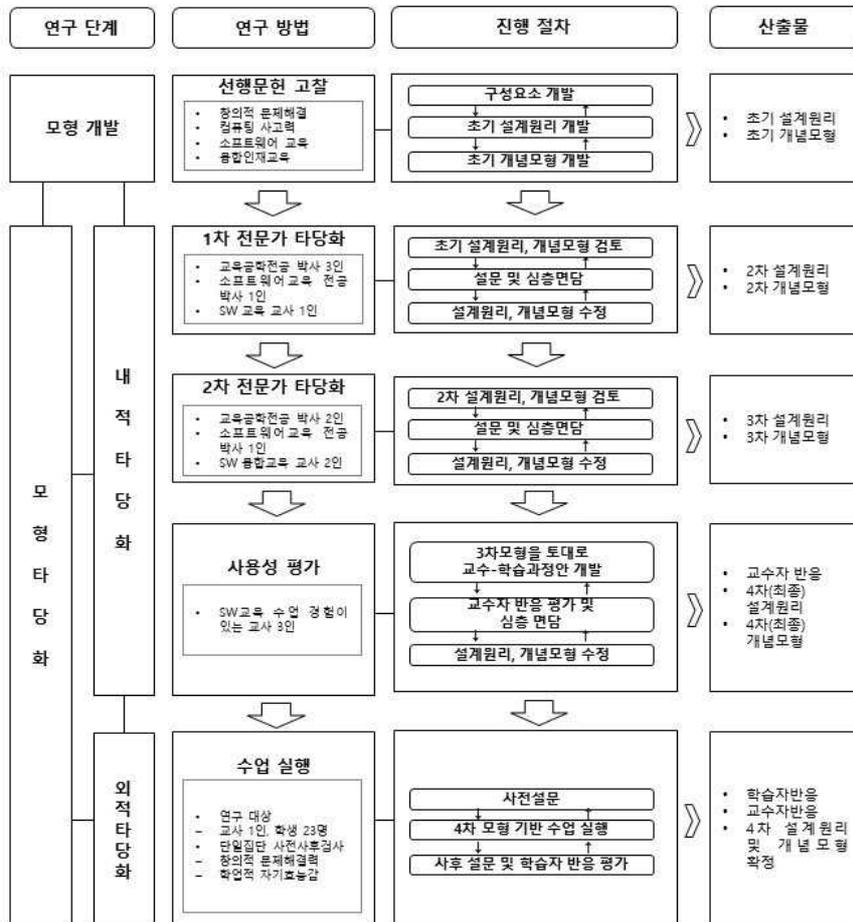
이 연구는 창의적 문제해결력을 위한 교과-소프트웨어융합 수업 설계 모형을 새롭게 개발하고 이를 타당화 하기 위한 연구로서, 이 목적을 달성하기 위한 모형연구 중에서 모형 개발 연구와 모형 타당화 연구의 두 단계를 수행하였다. 모형 개발 연구는 선행문헌 고찰을 통해 이루어졌고, 모형 타당화 연구는 전문가 타당화 검토 및 교수자의 사용성 평가를 통한 내적 타당화와 실제 학교 현장 적용 평가를 통한 외적 타당화의 두 단계로 진행되었다(<표 III-2> 참고).

<표 III-2> 본 연구의 단계별 주요 연구방법

Richey & Klein(2007)의 모형 연구		본 연구
단계	탐색주제	연구방법
모형 개발	<ul style="list-style-type: none"> <li>포괄적 모형 개발</li> <li>구성요소 과정 개발</li> </ul>	<ul style="list-style-type: none"> <li>선행문헌 검토</li> </ul>
모형 타당화	<ul style="list-style-type: none"> <li>모형 구성요소의 내적 타당화</li> </ul>	<ul style="list-style-type: none"> <li>전문가 타당화 검토 (1~2차)</li> <li>현장 교사의 사용성 평가</li> </ul>
	<ul style="list-style-type: none"> <li>모형 영향에 대한 외적 타당화</li> </ul>	<ul style="list-style-type: none"> <li>현장 평가                             <ul style="list-style-type: none"> <li>- 교수자 반응</li> <li>- 학습자 반응</li> </ul> </li> </ul>

본 연구의 세부적인 진행절차와 이에 따른 산출물은 다음과 같다. 우선 선행문헌 고찰을 통해 1차 수업 설계의 설계원리, 개념 모형, 절차 모

형을 개발하였다. 도출된 초기 모형에 대한 1차 전문가 타당화 검토를 거쳐 2차 수업 설계의 설계원리, 개념 모형, 절차 모형을 개발하였다. 2차 수업설계 모형과 수업설계 원리는 2차 전문가 타당화 검토를 통해 내적 타당도를 평가받고, 도출된 개선사항을 반영하여 3차 수업 설계의 설계원리, 개념모형, 절차모형을 개발하였다. 3차 수업설계 원리와 수업 설계모형은 현장 교사에게 사용성 평가를 받아 내적 타당도를 검증하였다. 이 후 외적 타당화를 위한 현장 평가를 실시하고, 학습자 반응과 교수자 반응을 확인하여 이를 사용성 평가 결과와 함께 수업설계 원리, 개념 모형, 절차 모형에 최종적으로 반영하여 수정 및 보완하여 완성하였다([그림 III-1] 참고).



[그림 III-1] 본 연구의 단계별 연구 방법, 진행절차 및 산출물

## 1. 내적 타당화

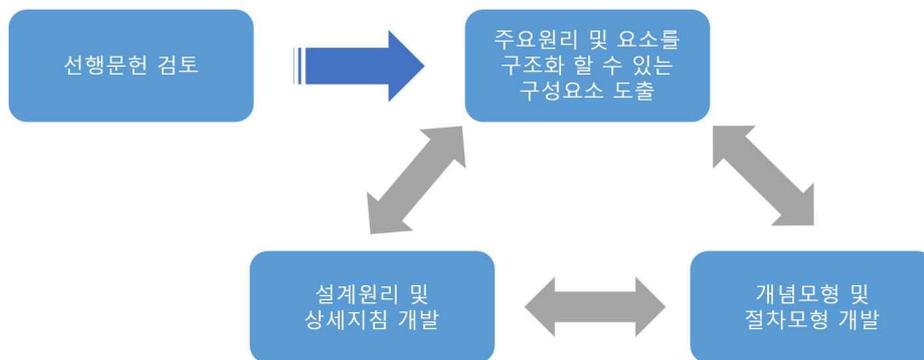
모형 개발을 위해서는 사례연구, 텔파이기법, 심층면담, 문헌검토, 조사연구, 소리내어 생각하기 등의 방법을 사용할 수 있다(Richey & Klein, 2007). 이 중에서 이 연구는 선행 문헌을 검토방법을 사용하여 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 위한 주요 설계원리를 확인하였다. 이후 설계원리를 개념적으로 구조화 할 수 있는 구성 요소를 도출한 다음, 설계원리와 상세지침, 개념모형, 절차모형을 개발하였다. 선행 문헌 검토를 통해 개발된 초기 모형과 설계 원리는 전문가 검토를 통해 타당성을 검토 받았다. 이후 실제 현장의 교사들을 통해 사용성 평가를 실시하여 수정 및 개선사항을 확인하였고, 이를 반영하여 수업설계 모형과 설계 원리를 지속적으로 수정·보완하였다.

### 가. 선행문헌 검토를 통한 초기 모형 개발

설계원리 및 모형을 개발하기 위한 가장 기초적이면서 핵심적인 방법은 선행문헌 검토이다. 본 연구는 선행문헌 검토를 통해 설계원리 및 모형을 개발하기 위해서 Reigeluth(1983)가 제안한 이론구성 절차(theory-construction process)를 따랐다. 그는 교수설계 이론을 정립하는 방법으로 귀납적 접근과 연역적 접근의 네 가지 단계를 제시하고 있다.(Reigeluth, 1983). 먼저 귀납적 접근의 첫 번째 단계로 각종 자료와 경험, 직관을 통해 ‘형성적 가설’을 마련하고, 두 번째 단계로 교수설계이론 또는 모형에서 중요하다고 여겨지는 변인들을 확인하고, 세 번째 단계로 원리들을 추출한 후 네 번째 단계로 교수설계 모형들과 이론들을 개발하는 것이다(Peterson, 2007). 특히 그는 모형 개발 시 귀납적 접근과 연역적 접근을 사용할 것을 강조하였는데, 귀납적 접근 방법은 위의 절차를 순서대로 따르는 것이고, 연역적 접근 방법은 귀납적 접근의 첫 번째 단계에서 세 번째 단계를 거치기 전에 네 번째 단계를 미리 개발하는 방식으로, 즉 모형을 먼저 개발한 후 원리를 도출하는 것이다.

Reigeluth는 이러한 구성 절차가 일반적으로 귀납적 접근과 연역적 접근이 함께 포함되어 이루어지는 순환적인 과정이라고 강조하였다.

본 연구는 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 모형과 원리를 개발하기 위하여 모형 개발을 위한 귀납적 접근과 연역적 접근을 모두 활용하였다. 즉, 초기 모형과 원리의 개발은 귀납적인 방법을 중심으로 선행문헌 검토를 통해 주요한 변인들을 확인하고 이를 개념적으로 구조화 할 수 있는 구성요소를 도출한 후, 이에 따라 설계원리를 개발하고 설계 모형을 개발하였다. 이후 초기 설계원리와 설계 모형을 수정하는 과정에서 연역적 접근 방식을 활용하여 선행문헌 검토를 통해 중요하다고 여겨지는 변인들을 구성요소로 도출한 후, 구성요소들 간의 관계를 나타내줄 수 있는 모형을 개발하고, 모형의 각 구성요소를 설계하기 위한 원리 및 상세 지침을 도출하는 과정을 거쳤다.



[그림 III-2] 선행문헌 검토를 통한 설계원리 및 모형 도출 과정

선행문헌은 국내외 저널에 게재된 논문, 학술대회 발표 논문, 주제와 관련된 단행본, 기관연구보고서, 기사, 웹사이트 등의 자료를 탐색하였다. 연구 주제별 키워드를 사용하여 서울대학교 중앙도서관 (<http://library.snu.ac.kr>)과 구글 학술검색([scholar.google.kr](http://scholar.google.kr)), 한국교육학술정보원이 제공하는 학술연구정보서비스(<http://www.riss.kr>)을 통해 적절한 문헌을 지속적으로 탐색하여 주요 연구 및 최신 문헌들을 검토하였다. 선행문헌 검토 시 관련 주제는 창의적 문제해결, 소프트웨어 교육, 융합교육을 대주제로 하고 소프트웨어교육 수업설계모형, 소프트웨어교

육 수업설계원리, 교과융합 소프트웨어교육 수업설계모형, 교과융합 소프트웨어교육 수업설계원리를 소주제로 분류하여 검토하였다. 덧붙여, 창의적 문제해결을 위한 교수설계모형, 소프트웨어교육 학습 환경, 융합교과 수업 등에 대한 문헌을 추가적으로 검토하였다. 이를 통해 창의적 문제해결력을 위한 교과-소프트웨어융합 수업설계를 위한 구성요소와 설계원리로 고려해야 하는 요인들을 확인하였다.

구성요소는 선행문헌 검토를 통해 확인된 내용들이 어떻게 구조화 될 수 있는지, 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계에서 중요하게 고려해야 할 변인들은 무엇인지를 중심으로 도출하였다. 설계원리는 선행 문헌에서 확인한 일반적인 원리들을 도출한 구성요소에 따라 내용과 수준을 고려하여 분류하고 재진술하여 개발하였다. 이 과정에서 원인과 결과를 드러내는 일반적인 명제들은 설계원리로, 그 원리를 어떻게 구현할 수 있는지 실제적인 행동지침을 내려주는 처방적인 내용은 상세 지침으로 구분하여 초기 설계 원리와 상세 지침을 개발하였다. 개념 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계와 관련한 요소는 무엇이며 이들 간의 관계는 어떠한지를 설명하기 위해 각각의 구성요소들을 시각적으로 나타내고 각 구성요소와 구성요소 간의 관계를 설명하도록 개발하였다. 절차 모형은 이 수업설계가 어떤 단계에 따라 설계되고 각 단계마다 중요하게 고려해야 할 요소와 설계활동은 무엇인지에 대해 전체적인 과정과 절차를 안내할 수 있도록 개발하였다.

## 나. 모형에 대한 전문가 타당화

선행문헌 검토를 통해 개발한 초기 설계모형과 설계원리의 내적 타당도를 검증하기 위하여 전문가 타당화 검토를 실시하였다. 전문가 타당화를 통해 초기 연구결과물의 타당성과 적절성을 검토하고 전문가들로부터 제안사항 및 개선 사항을 확인하였으며, 2차에 걸친 전문가 타당화를 통해 설계 원리와 개념모형, 절차 모형을 지속적으로 수정 및 개선하였다.

## 1) 연구 참여자

전문가 검토를 위해서는 해당 분야의 전문가를 선정하는 것이 중요한데, 이 연구에서는 Davis(1992)와 연은경(2013)이 제안하는 방법을 참고하여 전문가를 선정하였다. 연구 참여자는 해당 분야의 석·박사 학위를 소지하고, 연구주제와 관련한 논문을 게재하였거나 주제와 관련한 현업 경력이 7년 이상인 자를 선정하였다. 따라서 이 연구에서는 교육공학 박사학위 소지자로서 수업 설계모형과 원리 개발에 전문성을 지닌 교수설계 전문가와 교과융합 수업이나 소프트웨어를 활용한 수업 경험이 있는 현직 교사에게 개별적인 연락을 통해 설계원리 및 모형의 타당화 검토를 의뢰한 다음 연구에 참여 의사를 밝히고 참여에 동의한 전문가를 토대로 전문가 패널을 구성하여 검토 작업을 진행하였다. 이 연구에서 전문가 타당화에 참여한 전문가 프로필과 총 두 차례의 전문가 타당화 중에서 개인별 참여 검토 단계는 <표 III-3>와 같다.

<표 III-3> 전문가 패널 프로필 및 타당화 참여 단계

참여 전문가 프로필					타당화 참여	
구분	직업	경력	최종 학력	전문분야	1차	2차
A	초등교사	13년	박사	교수설계, 모바일교육	✓	
B	조교수	10년	박사	교수설계	✓	✓
C	연구원	18년	박사	교수설계, 멀티미디어 개발	✓	✓
D	중등교사	10년	석사	소프트웨어교육, 교과융합교육	✓	
E	겸임교수	20년	박사	소프트웨어교육, 교과융합교육	✓	
F	고등교사	13년	학사	소프트웨어교육,		✓

				교과융합교육		
G	조교수	25년	박사 수료	소프트웨어/AI교육, 지능형교육시스템		√
H	초등교사	7년	석사	STEAM수업설계, 테크놀로지 기반 수업		√

## 2) 연구도구

설계 원리와 설계 모형의 내적 타당성 확보를 위해 전문가 타당화를 실시하였고, 타당화 검사 도구를 개발하여 활용하였다. 설계 원리와 설계 모형에 대한 전문가 타당화는 1차, 2차에 걸쳐 총 두 차례로 이루어졌다. 1차 전문가 검토를 위한 검사지는 ‘1. 연구 소개’ 부분과 ‘2. 타당도 검토’의 두 부분으로 구성되었다. 연구 소개 부분은 1) 연구의 필요성과 목적, 2) 구성요소, 설계 원리 및 지침, 모형의 개발 및 도출 과정과 주요 내용으로 구성되었다. 타당도 검토 부분은 1) 도출된 구성요소에 대한 타당화 질문지와 2) 수업설계 개념모형에 대한 타당화 질문지, 3) 설계원리 전반에 대한 타당화 질문지, 4) 개별 설계원리에 대한 타당화 질문지, 5) 수업설계의 절차모형에 대한 타당화 질문지로 구성되었다. 타당화 질문지의 평가문항은 김선희(2014)와 김성욱(2016)이 사용한 문항을 본 연구에 맞게 수정하여 사용하였다. 타당화 검토 문항은 4점 척도에 따라 응답하도록 구성되었으며, 각 질문지의 마지막 부분에는 모형과 원리의 수정 및 보완을 위해 전문가가 자유롭게 의견을 제시할 수 있도록 개방형으로 제작된 전문가 의견란을 제공하였다.

<표 III-4>는 전문가 타당화에서 사용된 설계원리 및 모형 개발 과정에 대한 평가 문항이다.

<표 III-4> 모형 개발 과정에 대한 전문가 타당화 문항

평가 문항	
선행문헌 탐색의	• 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의

적절성	핵심 구성요소, 설계원리 및 지침, 개념모형, 절차모형을 도출하기 위해 관련 문헌의 탐색이 적절하게 이루어졌는가?
선행문헌 고찰 결과 요약 및 해석의 적절성	<ul style="list-style-type: none"> <li>• 선행연구는 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 핵심 구성요소, 설계원리 및 지침, 개념모형, 절차모형을 개발하기 위해 적절하게 요약되고 해석되었는가?</li> </ul>

<표 III-5>은 구성요소에 대한 전문가 타당화 평가 문항이다. 도출된 구성요소에 대한 타당화 평가는 구성요소의 적절성, 수준의 동일성, 하위 요소의 적절성, 설명내용의 이해 용이성 등에 관한 총 아홉 개의 문항으로 구성되었고 각 문항은 4점 척도에 따라 응답하도록 구성되었다.

<표 III-5> 구성요소에 대한 전문가 타당화 평가 문항

평가 문항	
	<ul style="list-style-type: none"> <li>• 도출된 구성요소는 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 시 고려해야 할 핵심 요소들로 타당하게 구성되었는가?</li> </ul>
	<ul style="list-style-type: none"> <li>• 도출된 구성요소는 동일한 수준의 요소(또는 용어)로 적절하게 구성되었는가?</li> </ul>
	<ul style="list-style-type: none"> <li>• 각 구성요소별로 제시된 하위요소는 적절하게 구성되었는가?</li> </ul>
	<ul style="list-style-type: none"> <li>• 구성요소별 설명내용은 적절하게 표현되었는가?               <ul style="list-style-type: none"> <li>· ‘컴퓨팅사고력 기반 창의적 융합설계활동 설계’에 대한 설명 내용</li> <li>· ‘교과융합 문제’에 대한 설명내용</li> <li>· ‘학습도구(창의적 사고도구, 컴퓨팅도구) 및 정보자원’에 대한 설명 내용</li> <li>· ‘공유 및 평가’에 대한 설명내용</li> <li>· ‘협력적 학습환경’에 대한 설명 내용</li> <li>· ‘안내’에 대한 설명 내용</li> </ul> </li> </ul>

<표 III-6>은 전문가 타당화에서 활용한 개념모형에 대한 타당화 평가 문항이다. 개념모형에 대한 타당화 문항과 설계원리 전반에 대한 타당화 문항, 절차 모형에 대한 타당화 문항은 선행연구(나일주, 정현미, 2001; 김선희, 2014; 김성욱, 2016)에서 사용했던 질문지를 참고하여 구성하였다. 선행 연구에서는 타당성을 검토하기 위하여 타당성, 설명력, 유용성, 보편성, 이해도의 총 다섯 영역에 대해 검토하였으며, 이 연구에서도 다섯 영역에 대하여 연구의 성격에 맞게 내용을 수정 및 보완한 후 평가 문항을 개발하였다. 각 항목은 4점 척도로 응답하도록 구성하였으며, 질문지의 마지막 부분에는 전문가가 자유롭게 자신의 의견을 제시할 수 있도록 개방형으로 제작된 전문가 의견란을 제공하였다. 이후 설계원리 전반과 절차모형에 대한 타당화 평가 문항은 개념 모형에 대한 타당화 평가 문항을 설계원리 전반과 절차모형에 대한 내용을 수정한 후 사용하였다.

<표 III-6> 개념모형에 대한 타당화 평가문항

영역	평가문항
타당성	<ul style="list-style-type: none"> <li>본 개념모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 참고할 수 있는 설계모형으로 타당하다.</li> </ul>
설명력	<ul style="list-style-type: none"> <li>본 개념모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 고려해야 할 요소들을 잘 설명하고 있다.</li> </ul>
유용성	<ul style="list-style-type: none"> <li>본 개념모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 유용하게 활용될 수 있다.</li> </ul>
보편성	<ul style="list-style-type: none"> <li>본 개념모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 보편적으로 적용할 수 있다.</li> </ul>
이해도	<ul style="list-style-type: none"> <li>본 개념모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는 데 필요한 변인들이 이해하기 쉽게 표현되었다.</li> </ul>

### 3) 연구진행 절차

창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형을 위한 전문가 검토를 진행하기 위해 먼저 이메일 또는 통화를 통해 전문가 타당화를 의뢰하였고, 연구참여 의사를 밝힌 전문가들을 전문가 패널로 구성하였다. 이후 각각의 전문가에게 연구의 핵심적인 내용을 설명하기 위해 전화통화를 통해 연구 내용의 전반적인 내용을 자세히 설명하고, 연구의 핵심적인 사항을 요약한 설명문을 타당화 질문지와 함께 첨부하여 이메일로 송부한 후, 응답을 회신하였다. 단, 소프트웨어 교육에 관련한 전문가는 교육공학을 전공한 교수설계 전문가와는 달리 교수설계 모형이나 설계 원리 도출에 대한 사전 이해가 필요하여 연구자와의 직접적인 만남을 통해 연구를 소개하였고, 타당도 진행방법에 대하여 자세히 안내한 후 타당도 검토를 진행하였다. 이 과정은 내용 전문가의 개인 이해도에 따라 약 한 시간 정도 소요되었다.

전문가 타당화를 통해 수집된 응답내용에 대한 타당도와 신뢰도를 확인하기 위해 내용 타당도 지수(content validity index: CVI)와 평가자간 일치도 지수(inter-rater agreement: IRA)를 사용하여 분석하였다. CVI는 각 설문 항목에 대한 전문가들의 평가가 어느 정도 일치하는지를 보여주는 것이다(Lynn, 1986; Rubio et al., 2003). 일반적으로 CVI 값이 .80 이상이면 타당하다고 해석한다. CVI 값은 각 설문항목에 대해 타당하다고 평가한 전문가의 수를 전체 전문가의 수로 나누어 구하게 되는데, 4점 척도(4: 매우 그렇다, 3: 그렇다, 2: 그렇지 않다, 1: 전혀 그렇지 않다)로 구성된 타당화 항목 중 전문가 평정 값이 3과 4일 경우에는 긍정적인 평가로, 1과 2는 부정적인 평가로 처리하여 결과를 분석한다.

IRA는 전문가들의 평가를 얼마나 신뢰할 수 있는가를 결정할 수 있는 지수이며, 평가자 간 동일하게 평가한 항목수를 전체 항목수로 나눈 값을 의미한다(Rubio et al., 2003). IRA값이 0이면 평가자 간의 의견이 완전히 불일치 한다는 것이고, IRA 값이 1이면 평가자 간의 의견이 완전히 일치한다고 판단할 수 있다. 일반적으로 IRA값이 .80이상이면 전문가

들의 평가를 신뢰할 수 있다고 한다(Rubio et al., 2003).

구성요소, 설계원리, 개념모형, 절차모형에 대한 수정 및 보완 사항을 묻는 개방형 문항에 대한 전문가 응답은 전문가가 직접 타당화 질문지에 서술식으로 기록하게 하였다. 면담 종료 후에는 전문가 검토 응답에 대한 분석 및 검토 과정을 거쳐 개선사항을 정리한 후 설계 모형 및 설계 원리에 반영하였다.

## 다. 사용성 평가

### 1) 참여자

사용성 평가는 이 연구에서 개발된 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형과 원리가 실제 학교 현장의 교사들에게 도움이 되는지를 확인하고 평가하기 위한 목적으로 실시되었다. 이를 위해 서울에 위치한 초등학교에 근무하고 있는 초등학교 교사 3인이 사용성 평가에 참여하였다. 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형과 원리가 교과목의 특성과 관계없이 일반적으로 활용될 수 있는 모형인지, 실제 설계 모형 적용에 어려움은 없는지, 교과목의 특성에 따라 어떤 변화와 차이점이 있는지를 확인하기 위하여 관심분야가 서로 다른 교사들로 구성하였다.

<표 III-7> 사용성 평가 참여자 프로필

현장교사	소속	경력	관심분야	성별
A	충남 ○○초등학교	7년	소프트웨어교육, 교수설계	남
B	서울 ○○초등학교	5년	사회과 교육, 디지털교과서	여
C	대구 ○○초등학교	7년	과학과 교육, STEAM	여

## 2) 연구도구

사용성 평가를 위해 사용된 연구도구는 ‘1. 연구의 소개’와 ‘2. 평가 질문지’ 부분으로 구성하였다. 연구 소개 부분은 1) 연구의 필요성과 목적, 2) 구성요소, 설계원리 및 지침, 모형의 개발 및 도출과정과 주요 내용으로 구성하였다. 평가 질문지 부분은 1) 사용성 평가 문항과 2) 개별 설계원리 및 상세지침의 타당도로 구성하였다.

사용성 평가 문항은 수업설계 원리, 개념 모형, 절차 모형이 각각 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계에 대한 이해와 설계과정, 실제 설계에 도움이 되는지에 관한 총 5문항을 4점 척도로 응답하도록 구성하였다. 또한 마지막 부분은 각각의 강점과 약점, 개선점, 설계하기 어렵거나 이해하기 어려운 부분, 개선이나 추가가 필요한 사항에 대하여 자유롭게 의견을 진술할 수 있도록 개방형 문항으로 구성하였다 ([부록 6] 참고). 개별 설계 원리와 상세 지침은 각각의 문항에 대하여 도움이 되었는지를 4점 척도로 응답하도록 구성하였으며, 설계 원리 및 상세지침에 추가되어야 할 요소와 강점, 약점, 개선점에 대한 의견을 자유롭게 제시할 수 있도록 하였다.

## 3) 연구 진행 절차

사용성 평가는 초등학교 현장에서 근무하고 있는 교사들 중 평소 수업에 소프트웨어를 활용한 경험이 있거나 이를 활용할 계획을 가지고 있는 교사들을 섭외하였다. 연구자는 지인들을 통해 소개를 받아 사용성 평가 대상인 교사를 선정하였으며, 이후 개별적으로 연락을 하여 연구의 핵심적인 내용과 사용성 평가 방식 등에 대한 전반적인 설명을 한 후, 연구에 참여하기로 동의한 세 명의 현장 교사에게 연구도구를 이메일로 송부하였다. 이후, 연구자와 현장교사의 일대일 면담을 통해 연구도구의 구체적인 활용방법 및 수업 설계방법에 대해 설명하고 이해되지 않거나 평가하는 방법 등에 관하여 문답하였다. 일대일 면담은 현장교사에 따라 약

30분 가량 온라인 화상회의 프로그램을 통해 진행되었다.

연구자의 설명을 들은 교사는 각자 자신의 상황에 따라 편리한 시간에 제시된 수업설계 원리, 개념 모형, 절차 모형에 따라 수업을 설계하였으며, 수업 설계를 마친 후 평가 질문지에 응답하여 이메일을 통해 연구자에게 전달해 주었다. 이러한 과정은 교사 개인별로 약 일주일에서 이주일 정도 소요되었으며, 사용성 평가를 통해 확인된 사항은 이후 현장평가 결과에서 확인되는 사항과 함께 최종 수업설계 원리와 모형의 수정에 반영하였다.

## 2. 외적 타당화

외적 타당화 연구(external validation study)는 설계 및 개발 모형을 사용하여 만들어진 산출물의 영향을 경험적으로 검증하는 것이다(Richey & Klein, 2007). 따라서 모형에 따라 교수적 산출물을 만들고 이 산출물이 학습자, 의뢰인, 조직에 실제로 어떤 영향을 주는지를 탐색한다(Richey & Klein, 2007). 본 연구에서 수업설계 모형의 외적 타당화를 위한 현장 평가는 현장 교사가 본 연구에서 도출된 설계 원리, 개념 모형, 절차 모형을 토대로 직접 수업을 설계하고 실제 수업 현장에서 실행한 후, 교수자와 학습자의 반응을 확인하여 모형의 효과성을 검토하는 것이다. 본 연구에서는 실제 수업을 진행한 교수자와 학습자의 반응을 분석하여 설계 원리, 개념 모형, 절차 모형을 추가적으로 수정 및 보완하였고, 학습자들의 창의적 문제해결력, 학업적 자기효능감에 대한 설계 모형의 효과를 확인하였다.

### 가. 연구 참여자

모형의 외적 타당화를 위하여 학교 수업 시간에 크롬북 및 태블릿 PC를 개인별로 사용할 수 있고, 와이파이 등 네트워크 인프라가 갖추어져 교과-소프트웨어융합 수업을 실행할 수 있는 경기도 소재 U 초등학교 1

개 학급 27명과 담임교사 1명이 참여하였다(<표 III-8> 참조). 외적 타당화에 참여한 담임 교사와 일부 학습자들은 이전에 소프트웨어를 수업 및 방과 후 활동에서 사용한 경험이 있다. 현장 평가를 초등학교 학급을 대상으로 선정한 이유는 초등학교가 중·고등학교에 비하여 교육과정 재구성 여건이 잘 마련되어 있고, 교수자도 여러 교과를 진행하기 때문에 본 연구에서 도출한 교과-소프트웨어수업 모형의 실제 학급 현장 적용이 용이하기 때문이다. 또한 실과 교과에서 소프트웨어를 배우기 시작하는 초등학교 고학년 학생을 대상으로 선정하였다.

<표 III-8> 현장평가 연구 참여자

교사 참여자(1인)	담임교사, 경력 10년	수업 시수	수업 일시
학생 참여자(23인)	6학년 1개반 (연구참여 동의 23인)	4	2020. 12.22 ~2020.12.28

본 연구는 참여 대상자가 미성년자이기 때문에 학생 개인과 함께 법정 대리인(학부모)에게 서면 및 온라인 문서를 통해 연구 참여 동의를 받고 자료를 수집하였다. 해당 학급 27명의 학생 중 연구 참여 의사를 밝힌 23명이 대상으로 선정되었다. 연구 참여 동의서는 참여 대상자의 익명성과 사생활 보호 및 비밀보장에 관한 내용이 포함되어 있으며 참여를 원치 않을 시 언제라도 중단할 수 있음이 안내되었다. 연구를 통해 수집된 모든 자료는 연구 목적 이외의 다른 용도로 사용하지 않을 것이며 자료에 대한 비밀을 지키기 위해 연구자가 직접 자료를 잠금 처리한 곳에 보관하고 암호가 설정된 개인 컴퓨터에서 분석한 후 폐기할 것을 약속하였다.

#### 나. 현장적용 절차 및 자료 분석 방법

연구자는 연구참여 의사를 밝힌 교사와 두 차례의 사전 회의를 가졌

다. 첫 번째 사전 회의에서는 연구자가 참여 교사에게 본 연구의 방향과 전반적인 내용에 대해 소개를 하였다. 사전 회의 전 연구자는 연구 내용과 수업설계 모형 및 원리가 제시된 자료를 이메일로 공유하였고 참여 교사가 이를 검토하여 궁금하거나 확인이 필요한 사항에 대해 연구자와 질의응답 하는 형식으로 진행되었다. 두 번째 사전 회의에서는 구체적인 연구 절차가 안내되었으며, 교과-소프트웨어융합 수업을 적용한 교과목과 단원, 수업 내용에 대해 협의하였다. 이후 교사는 연구자가 제공한 설계 원리와 설계 모형에 따라 직접 수업을 설계한 후 본 수업을 진행하였다.

본 수업은 총 6차시로 구성되었으나, 학교 일정 상 수업을 나흘에 걸쳐 1~2차시, 3차시, 4~5차시, 6차시로 모듈형 수업을 진행되어 실질적으로 4차시의 수업으로 진행하게 되었다. 특히 외적 타당화 수업의 실행 직전 코로나바이러스감염증-19로 인한 사회적 거리두기가 3단계로 격상되어 오프라인 수업이 불가능한 상황으로 인해 모든 수업을 온라인 상에서 화상회의 소프트웨어를 활용하여 진행하게 되었다. 연구자는 온라인 화상회의 소프트웨어를 통해 모든 수업에 참여하여 모니터링 하였다. 본 수업에 들어가기 전에 창의적 문제해결력, 학업적 자기효능감에 대한 사전 검사를 실시하였고, 전체 수업을 마친 후에는 동일 항목에 대한 사후 검사를 실시하여 단일집단 사전사후 검사를 진행하였다.

질적 연구의 경우 수업 관찰과 면담을 실시하였다. 수업 관찰은 수업이 진행되는 과정에서 학습자가 어떻게 교과-소프트웨어융합 수업을 통해 창의적으로 문제를 해결해나가는지 조사하기 위해 실시하였다. 연구자는 화상회의 소프트웨어(ZOOM)를 활용하여 이루어지는 비대면 수업에 전 차시 참여하였으며, 수업에 방해가 되지 않도록 음소거 상태를 유지하고 영상 녹화를 진행하였다. 만약 영상 녹화가 불편함을 유발할 경우 녹화를 중지할 것이라는 점을 학생들에게 수업 전에 충분히 설명하고 녹화를 진행하였다.

수업 후 5명의 학생을 대상으로 심층 면담을 실시하였다. 2명과 3명씩 두 집단을 구성하여 각 집단별로 1회, 약 30분씩 심층 면담을 실시하였

다. 면담은 화상회의 소프트웨어를 통해 녹화한 후 전사하여 분석에 활용하였다.

## 다. 검사 도구

### 1) 교수자 반응 평가

창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형에 대한 교수자의 반응을 확인하기 위하여 평가 질문지를 개발하였다. 교수자의 반응 평가 질문지는 모형에 대한 사용성 평가 문항과 개별 설계원리 및 상세지침에 대한 의견을 묻는 문항으로 구성되었다.

모형에 대한 사용성 평가 문항은 앞서 진행된 현장 교사의 사용성 평가 문항과 동일한 문항을 사용하였다. 교과-소프트웨어융합 수업설계의 설계원리, 개념모형, 절차모형에 대한 이해와 실제 설계에 도움이 되는지에 대한 문항을 4점 척도로 응답하도록 평가 문항을 개발하였으며, 강점과 약점, 개선점에 대한 개방형 문항을 통해 자유롭게 진술하도록 구성하였다. 개별 설계원리와 상세지침에 대한 의견을 묻는 문항은 개별 설계원리와 상세지침이 수업 설계에 어느 정도 도움이 되었는지에 대한 사항을 묻는 것으로 4점 척도로 응답하게 구성되었으며, 개방형 문항을 통해 강점과 약점, 개선점에 대한 의견을 자유롭게 추가할 수 있도록 하였다.

### 2) 학습자 반응 평가

모형의 외적 타당화를 위한 학습자 반응 평가를 위한 문항은 크게 세 부분으로 구성하였다. 첫 번째 부분은 수업 전후로 학습자들의 창의적 문제해결력, 학업적 자기효능감을 확인하는 것이고, 두 번째 부분은 수업 차시별 학습자 활동 결과를 확인하고 분석하는 것이며, 세 번째 부분은 반구조화된 면담을 통해 수업에 대한 의견 및 효과에 대한 학습자들의

의견을 수합하고 분석하는 것이다.

본 연구에서 도출된 수업설계 원리와 개념모형, 절차모형을 활용하여 설계된 수업이 학습자들의 창의적 문제해결력과 학업적 자기효능감에 미치는 영향을 확인하기 위하여 [부록 8]과 같은 검사 도구를 사용하였다. 창의적 문제해결력은 본 수업설계모형이 개발하고자 하는 역량으로 교과-소프트웨어융합 수업을 통해 개발된 학습자의 지식, 사고기능, 기술의 이해 및 확산적·비판적 사고를 확인하기 위해 측정하였다. 또한 학업적 자기효능감은 학습자들이 과제 수행을 위해 필요한 수업 활동을 조직하고 실행해 나가는 자신의 능력에 대해 내리는 판단을 확인하기 위해 측정하였다. 특히 학업적 자기효능감은 학습자의 학습계획, 학습실행, 학습평가 수준에 긍정적인 영향을 미치며, 자신감과 인내력, 도전적인 문제의식, 학습 만족도 등의 향상을 위한 핵심 요인이라는 선행연구(주영주, 박수영, 2015; 차현진, 김민하, 2020; 최희진, 2016)를 토대로 이를 확인하기 위해 측정하였다.

창의적 문제해결력은 한국교육개발원(2001)에서 발간한 ‘간편 창의적 문제해결력 검사 개발 연구’를 기반으로 2004년 서울대학교 심리연구실 MI연구팀에서 개발한 문항으로 전용주(2017)의 연구에서 사용된 문항을 사용하였다. 전용주(2017)의 연구에서 사용된 창의적 문제해결력 검사지의 신뢰도 계수 Chronbach's  $\alpha$ 는 .97이었으며, 본 연구에서 나타난 Chronbach's  $\alpha$ 는 .92로 양호한 것으로 나타났다. 학업적 자기효능감 확인을 위해서는 김아영, 박인영(2001)의 학업적 자기효능감 척도의 문항을 사용하였다. 학업적 자기효능감 설문문항은 총 28문항으로 각각 과제난이도 선호, 자기조절 효능감, 자신감의 하위 항목으로 나누어져 있으며, 각 항목의 Chronbach's  $\alpha$ 는 .84, .76, .74으로 대체로 양호한 것으로 나타났다. 본 모형의 수업 적용에 있어 측정은 수업의 전과 후에 각각 실시되었고, SPSS 18.0 프로그램을 사용하여 대응표본 t 검정을 실시하였다.

모형의 효과성을 확인하기 위해 학생들의 수업 산출물을 확인하고 분석하였다. 이를 위해 수업 차시별로 패들렛을 활용한 아이디어 도출, 구

글독스를 활용한 스토리보드, 엔트리를 활용한 홍보 애니메이션 제작 활동의 산출물을 확인하였다. 원격화상회의 프로그램인 줌(ZOOM)을 활용하여 산출물이 제작되는 과정 및 모듈별 활동 장면을 영상으로 녹화하였으며, 패들렛, 구글독스, 엔트리는 이미지 캡처 및 파일 다운로드를 통해 산출물을 확인하였다.

또한 교사가 추천한 학습자 5명에 대해 반구조화된 면담을 실시하였다. 면담은 교과-소프트웨어융합 수업에 대한 경험 여부와 장점 및 어려운 점, 수업 활동 중 가장 기억에 남는 활동과 이후 교과-소프트웨어융합 수업에 대한 학습 희망 여부에 대한 의견을 묻는 문항을 바탕으로 이루어 졌으며, 면담 내용은 동의하에 화상회의 소프트웨어 프로그램을 통해 녹화하여 분석에 활용하였다. 면담 중 학습자의 답변 내용은 면담 중간 중간 연구자가 종합하여 답변한 내용이 맞는지 여부를 묻고 확인하는 절차를 거쳤다. 인터뷰 내용 분석을 위해 인터뷰 전사 자료를 여러 번 반복하여 읽으며 연구자의 편견을 배제하고 아이디어를 기록 후, 연구와 관련성이 높은 내용을 중심으로 키워드를 도출하고 이를 질문지를 바탕으로 주제별로 범주화 하였다. 이후 주제별로 키워드를 포함한 전사내용을 범주화 하여 재정리하였고, 본 논문의 연구결과에 기술하였다.

## IV. 연구 결과

본 연구는 창의적 문제해결을 위한 교과융합 소프트웨어 교육을 위한 수업설계모형을 개발하는 것을 목적으로 한다. 본 연구는 총 세 단계의 과정을 거쳐 진행되었다. 첫째, 컴퓨팅 사고력과 융합 교육, 소프트웨어 교육에 관한 선행문헌 검토를 통해 초기 수업설계 모형 및 원리를 도출하였다. 둘째, 도출된 초기 수업설계 모형 및 원리에 대하여 현직 교사와 교육공학 전문가에게 전문가 타당화를 받는 내적 타당화 과정을 거치고, 이를 반영하여 수정·보완된 수업설계 모형 및 원리를 도출하였다. 셋째, 개발된 원리와 모형을 실제 수업 현장에 적용하여 모형의 효과성을 검증하고 교수자와 학습자의 반응을 살피는 외적 타당화 과정을 거친 후, 개선사항을 반영하여 최종 모형을 도출하였다.

### 1. 초기 모형 개발

#### 가. 모형의 구성요소 및 설계원리 도출

창의적 문제해결을 위한 교과융합 소프트웨어 수업설계의 이론적 구성요소를 도출하기 위하여 관련 선행문헌을 검토하였다. 검토 대상 문헌을 선정하기 위해 논문검색 사이트인 학술연구정보서비스(<http://www.riss.kr>), 구글 학술검색([scholar.google.kr](http://scholar.google.kr))와 서울대학교 중앙도서관(<http://library.snu.ac.kr>)에서 주요 키워드 검색을 통해 문헌을 검색하였다. 주요 키워드는 ‘창의성’, ‘창의적 문제해결’, ‘CPS 모형’, ‘소프트웨어 수업’, ‘컴퓨팅 사고력’, ‘코딩’, ‘프로그래밍’, ‘융합인재교육(STEAM)’, ‘교과융합’, ‘융합 수업’ 등을 사용하였다. 검색결과 중 교육 프로그램, 수업모형, 수업 설계원리 및 수업설계 모형 등에 대한 논문을 선정하여 검토하였다.

이상의 영역들에서 선행문헌을 검토하여 창의적 문제해결을 위한 교

과-소프트웨어융합 수업 설계를 위해 활용될 수 있는 요소들과 일반적인 설계 원리들을 도출한 후, 유사하거나 공통된 핵심 내용을 중심으로 유목화하고 통합하여 이를 구조화 할 수 있는 구성요소를 도출하였다. 결과적으로 창의적 문제해결을 위한 교과융합 소프트웨어 수업설계 모형의 초기 이론적 구성요소로 컴퓨팅 사고력 활용 창의적 융합설계활동, 교과융합문제, 학습도구 및 정보자원, 협력적 학습환경, 평가 및 공유 지원, 안내 제공의 여섯 가지 요소가 도출되었다(<표 IV-1> 참조).

<표 IV-1> 선행문헌 검토를 통한 일반적인 원리 확인 및 구성요소 도출

일반 설계원리 및 출처	문헌영역	공통내용	구성요소
학습자들에게 친숙하고 간단한 문제 상황을 제시하여 해결하도록 한다(구양미, 김영수, 노선숙, 조성민, 2006).	소프트웨어 창의적 문제 해결		
일상생활에서 발생하는 다양한 예시를 제시하고 토론할 수 있도록 지원한다(김수환, 2015; 송해덕, 2007; 손영수, 이광재, 2016).	소프트웨어 교과융합		
사전에 학습한 지식에 대한 실질적인 실습과정을 제공한다(손영수, 이광재, 2016).	소프트웨어	실생활의 친숙한 주제를 선정하고 학습활동을 해야 함	교과융합 문제
실생활의 맥락화된 비구조화 문제를 제시한다(강정찬, 2015; 김재휘, 김동호, 2016; 송해덕, 2007).	소프트웨어 교과융합		
스토리텔링 형태로 하나의 큰 주제 내에 발생할 수 있는 상황을 소재로 문항을 개발한다(김민정, 이원규, 김자미, 2017; Froyd, 2008;	소프트웨어 교과융합		

Shen, Jiang, & Liu, 2015).		
<p>학습 내용, 학습 환경, 학습자 수준, 학생들의 이해도, 학습 차시 등을 고려하여 중복된 내용을 축소하고, 어려운 내용을 심화한다(서경혜, 2009; 손영수, 이광재, 2016; 임화영, 윤희경, 2019).</p>	교과융합	
<p>융합하려는 교과들의 성취 기준을 확인 및 분류하여 이를 중심으로 주제를 선정하고 프로젝트를 구성한다(김수환, 2015; 임화영, 윤희경, 2019).</p>	소프트웨어 교과융합	<p>다양한 교과간 융합을 위해 내용을 재구성해야함</p>
<p>컴퓨팅 사고력 및 창의적 문제해결과 관련되는 여러 교과와 융합하여 내용을 재구성한다(손영수, 이광재, 2016; Weinberg, J. B. et al., 2001).</p>	소프트웨어 교과융합	
<p>수집된 데이터 및 데이터 패턴을 학생 스스로 분석하고 탐구할 수 있는 기회를 제공한다(변현정, 2019; 전수진, 2017; Seoane-Pardo, 2016).</p>	소프트웨어	
<p>실습을 하는 과정에서 학습자의 자율적 참여가 가능하도록 지원한다(구덕희, 우석준, 2018; 손영수, 이광재, 2016).</p>	소프트웨어	<p>학습자가 자율적, 주도적으로 학습에 참여해야함</p>
<p>학습자가 산출물을 직접 설계하고 만들어 볼 수 있</p>	소프트웨어	<p>CT활용 창의적 융합설계활동</p>

<p>도록 지원한다(이영기, 손창호, 2016).</p>		
<p>학습자에게 흥미롭고 의미 있는 학습 과제와 도구(어플리케이션, 프로그래밍 도구 등)를 제공한다(Kong, 2016).</p>	<p>소프트웨어</p>	
<p>문제해결에 대한 동기유발을 위해 다양한 발문과 자료를 활용한다(배학진, 이은경, 이영준, 2009).</p>	<p>창의적 문제 해결 교과융합</p>	<p>학습자의 흥미와 동기 유발을 위한 학습활동을 지원해야 함</p>
<p>학습자에게 외재적, 내재적 보상체계를 제공하여 새로운 문제에 대한 도전의식을 촉진시킨다(박현주 외, 2012; 한국과학창의재단, 2012).</p>		
<p>코딩 명령어 쿼북, 명령어 카드와 같은 형태의 교수 학습 자료를 지원한다(김수환, 2015).</p>	<p>소프트웨어</p>	
<p>언어화, 시각화 도구(마인드맵, 스프레드 시트 등)를 활용하여 문제의 추상화 과정을 지원한다(김성일, 2012; Repenning, Basawapatna, &amp; Escherle, 2016).</p>	<p>소프트웨어 창의적 문제 해결</p>	<p>학습자가 이해한 바를 시각적으로 표현할 수 있도록 지원해야 함</p>
<p>산출물을 시각적으로 디스플레이 할 수 있도록 지원한다(Cross, Bartley, Hamner &amp; Nourbakhsh, 2013; Hambruch, Hoffmann, Korb, &amp; Hosking, 2009).</p>	<p>소프트웨어</p>	

<p>프로젝트를 테스트하고 디버깅 하기 위한 예시를 제공한다(Brennan &amp; Resnick 2012; Cross et al., 2013; Wing, 2006).</p>	소프트웨어	<p>학습자가 점진적, 반복적으로 프로젝트를 설계하고 실행할 수 있어야 함</p>	
<p>다른 학습자나 교수자가 공유한 코드를 재사용하여 학습자가 자신만의 프로젝트를 완성할 수 있도록 지원한다(Brennan &amp; Resnick 2012; Kong, 2016).</p>	소프트웨어	<p>학습자가 점진적, 반복적으로 프로젝트를 설계하고 실행할 수 있어야 함</p>	
<p>다른 사람의 프로그래밍 코드를 재사용하여 새롭고 복잡한 산출물을 만들 수 있도록(리믹스) 지원한다(Kong, 2016).</p>	소프트웨어	<p>학습자가 점진적, 반복적으로 프로젝트를 설계하고 실행할 수 있어야 함</p>	
<p>학습자에게 적절한 난이도의 프로그래밍 언어를 선택하여 제공한다(손영수, 이광재, 2016; 최숙영, 2016; Cross et al., 2013; Quigley, &amp; Jamil, 2017; Taljaard, 2016; Weintrop et al., 2016).</p>	소프트웨어	<p>학습자가 효과적으로 활용할 수 있는 컴퓨팅 도구를 선정해야 함</p>	학습도구
<p>효과적이고 능률적으로 활용 가능한 컴퓨팅 도구를 선정한다(손영수, 이광재, 2016).</p>	소프트웨어	<p>학습자가 효과적으로 활용할 수 있는 컴퓨팅 도구를 선정해야 함</p>	학습도구
<p>학습자에게 해당 시스템과 사고 도구에 대한 사전 안내를 실시한다(임철일 외, 2014; Firertien, 1982).</p>	소프트웨어	<p>학습자가 효과적으로 활용할 수 있는 컴퓨팅 도구를 선정해야 함</p>	학습도구
<p>컴퓨팅 도구를 간단하고</p>	소프트웨어	<p>학습자가 효과적으로 활용할 수 있는 컴퓨팅 도구를 선정해야 함</p>	학습도구

<p>쉬운 수준(초보자용 도구)에서 점진적으로 복잡하고 복합적인 수준(전문가용 도구)으로 제공한다(Cross et al., 2013; Krauss &amp; Prottzman, 2017; Kong, 2016).</p>		
<p>발산적 사고도구를 활용하여 다양한 아이디어를 구상하고 도출할 수 있도록 지원한다(김성일, 2012; 노지예, 이정민, 2018; 전용주, 2017; 박태정, 2018; Firertien, 1982).</p>	<p>창의적 문제 해결</p>	
<p>수렴적 사고도구를 활용하여 문제에 대한 적절한 해결책을 찾을 수 있도록 지원한다(김성일, 2012; 노지예, 이정민, 2018; 전용주, 2017; 박태정, 2018; Firertien, 1982).</p>	<p>창의적 문제 해결</p>	<p>학습자가 발산·수렴적 사고를 효과적으로 활용할 수 있도록 지원해야 함</p>
<p>발산적 사고전략과 수렴적 사고전략을 균형있게 사용할 수 있도록 조언과 피드백을 제공한다(임철일 외, 2014).</p>	<p>창의적 문제 해결</p>	
<p>프로그래밍 활동을 위해 다양한 교수학습자료(코딩 명령어 쿼북, 명령어 카드)를 제공한다(김수환, 2015).</p>	<p>소프트웨어</p>	<p>학습활동을 위한 학습자원과 학습 환경을 조성해야 함</p>
<p>다양한 정보에 접근하고 활용할 수 있도록 지원한다(구양미 외, 2006; 김성기, 1993; 변현정, 2019; 임철일</p>	<p>소프트웨어 교과융합 창의적 문제</p>	

외, 2014).	해결		
온라인 웹기반 시스템을 활용하여 창의적 문제해결 활동을 할 수 있도록 안내한다(구양미 외, 2006; 임철일, 윤순경, 박경선, 홍미영, 2009; 임철일, 홍미영, 이선희, 2011; 홍미영, 2012; 홍현미, 2017; Chen & Cheng, 2009; Minamino & Kinoshita, 2010).	소프트웨어 창의적 문제 해결		
프로그래밍을 위한 팀을 다양한 구성원으로 구성하여 학습 활동을 할 수 있도록 지원한다.(임철일 외, 2014; Bacon, Stewart, & Anderson, 2001; Froyd, 2008).	소프트웨어 교과융합 창의적 문제 해결	학습자간 협력과 상호작용이 원활히 이루어질 수 있도록 지원해야 함	
협력과 팀워크가 촉진될 수 있도록 다양한 활동을 구성한다(Kong, 2016).	소프트웨어		
교수자는 학습 코치로서 그룹활동을 촉진한다(임철일 외, 2014; Lavonen, Autio, & Meisalo, 2004).	교과융합 창의적 문제 해결		협력적 학습환경
학습자의 자유로운 사고를 촉진하고 학습자간 의견을 즐겁게 공유할 수 있는 열린 교실 분위기를 형성한다(Hamza, & Griffith, 2006; DeSchryver, & Yadav, 2015; 박태정, 2018).	소프트웨어 교과융합 창의적 문제 해결	학습자간 자유롭게 의견을 교환할 수 있는 교실 문화를 형성해야 함	
문제해결에 대해 긍정적이	창의적 문제		

고 개방적인 태도를 가질 수 있도록 지원한다(송해덕, 2007).	해결		
개발내용을 공유하고 평가하여 학습자들 간 피드백과 성찰이 가능하도록 지원한다(구덕희, 우석준, 2018; 김재휘, 김동호, 2016; 변현정, 2019; Krauss & Prottsman, 2017).	소프트웨어 창의적 문제 해결 교과융합		
피드백을 받은 사항들을 기존 산출물에 반영시켜 개선시킬 수 있도록 지원한다(김재휘, 김동호, 2016; Seechaliao, 2017).	소프트웨어	산출물을 공유하고 피드백과 평가가 이루어지도록 해야 함	공유 및 평가
학습 과정 및 학습 활동의 다양한 지점을 다각적으로 평가한다(김재휘, 김동호, 2016; 박태정, 2018; 변현정, 2019; 한국과학창의재단, 2012; Brennan & Resnick, 2012; Lavonen et al., 2004).	소프트웨어 창의적 문제 해결 교과융합		
구체적인 방법을 알려주기보다 제안의 형태로 피드백을 제공한다(DeSchryver & Yadav, 2015).	창의적 문제 해결		
학습자에게 해당 시스템과 사고도구에 대한 사전 안내를 실시한다(김혜란, 최선영, 2019; 임철일 외, 2009; Froyd, 2008; Seoane-Pardo, 2016).	소프트웨어 창의적 문제 해결 교과융합	학습자에게 수업의 운영과 내용, 활용 도구, 평가 등에 대한 안내를 제공해야 함	안내
탐프로젝트의 안내 및 평가 계획을 홈페이지에 구	소프트웨어 창의적 문제		

체적으로 공지한다(배학진, 해결  
이은경, 이영준, 2009). 교과융합

학습자가 수업을 하는  
과정에서 진행할 내용 소프트웨어  
및 운영방식을 사전에 창의적 문제  
안내한다(김혜란, 최선영, 해결  
2019; 임철일 외, 2009; Froyd, 교과융합  
2008; Seoane-Pardo, 2016).

초기 모형의 구성요소에 따른 설계원리 및 상세지침은 선행문헌 고찰 결과 도출된 설계원리 및 상세지침을 유사한 내용을 중심으로 재정리 및 유목화하고 통합한 후 공통된 핵심 내용을 중심으로 범주화 하고, 이를 포괄할 수 있는 상위 내용을 초기 설계원리로 개발하였다. 또한 연구목적에 맞게 문장을 재진술 하면서 ‘설계원리’와 ‘상세지침’으로 수준을 분류하여 초기 설계 원리를 개발하였다. 초기 설계원리는 여섯 가지 구성요소를 중심으로 13개의 설계원리와 32개의 상세지침으로 구성되었다(<표 IV-2> 참조).

<표 IV-2> 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계를 위한 초기 설계원리 및 상세 지침

구성 요소	원리
컴퓨팅 사고력 (CT) 활용 창의적 융합설계 활동	<b>1. 자기주도학습 제공의 원리</b>
	스스로 능동적인 지식을 구성할 수 있는 학습경험 및 기회를 제공한다.
	1.1 친숙한 주제에 대한 토론과 발표를 통해 학습자가 스스로 문제를 설정, 계획하고 해결할 수 있도록 지원한다(강정찬, 2015; 김성호, 김신용, 강성미, 김용성, 2011; 김재휘, 김동호, 2016; 박현주 외, 2012; Weininger, 1977).
	1.2 수집된 데이터 및 데이터의 패턴을 학생 스스로 분석하고 탐구할 수 있는 기회를 제공한다(전수진, 2017).

<p><b>2. 감성적 체험 지원의 원리</b>          학습자가 성취, 실패, 노력 등 다양한 감성적 체험을 체험할 수 있도록 지원한다.</p>
<p>2.1 학습자가 수업에 몰입하고 흥미와 동기를 가질 수 있도록 다양한 학습활동(경험, 관찰, 실습, 토의, 설계 등)을 구성한다(박영석 외, 2013; 한국과학창의재단, 2012).</p>
<p>2.2 문제해결에 대한 동기 유발을 위해 다양한 발문과 자료를 활용한다(배학진 외, 2009).</p>
<p>2.3 학습자의 성취와 새로운 문제에 대한 도전의식을 촉진하기 위해 문제해결 이 후 내재적·외재적 보상체계를 마련한다(강호감, 김태훈, 2014; 박현주 외, 2012; 한국과학창의재단, 2012).</p>
<p><b>3. 시각화 지원의 원리</b>          교과 내용 및 컴퓨팅 원리에 대해 이해한 바를 효과적으로 시각화 할 수 있도록 지원한다.</p>
<p>3.1 문제에 대한 해결 아이디어 및 사고과정을 순서대로 조직화할 수 있도록 순서도 및 사고과정 맵의 활용에 대한 내용을 안내한다(이재호, 하희정, 2016; 전용주, 2017; Weintrop et al., 2016).</p>
<p>3.2 학습 산출물이 시각적으로 표현될 수 있도록 온라인 플랫폼을 지원한다(Cross et al., 2013; Kong, 2016).</p>
<p><b>4. 점진·반복성의 원리</b>          학습자가 점진적·반복적으로 아이디어와 문제 해결 방법을 개발할 수 있도록 지원한다.</p>
<p>4.1 학습자가 점진적, 반복적으로 프로젝트를 설계하고 실행할 수 있도록 피드백을 제공한다(Brennan, &amp; Resnick, 2012; Cross et al., 2013; Wing, 2006).</p>

	<p>4.2 창의적 융합설계 활동의 단계별로 발산적 사고와 수렴적 사고를 반복적으로 사용할 수 있도록 사고지원도구의 활용방법을 안내한다(구양미 외, 2006; 임철일 외, 2014; 홍미영, 2012).</p>
교과융합 문제	<p><b>5. 맥락 친숙성의 원리</b> 학습자에게 친숙한 일상생활 맥락의 과제와 예시를 제공한다.</p>
	<p>5.1 일상생활에서 일어나는 친숙하고 실제적인 과제와 사례를 다양하게 제시한다(구양미 외, 2006; 김수환, 2015; Cross, J. et al., 2013; Krauss &amp; Prottzman, 2017).</p>
	<p>5.2 시나리오 형태로 하나의 큰 주제 내에 발생할 수 있는 상황을 소재로 문제를 제시한다(김민정, 이원규, 김자미, 2017; Froyd, 2008; Shen et al., 2015).</p>
	<p>5.3 실생활의 문제처럼 복잡하고 답이 하나 이상인 주제나 과제, 문제를 비구조적으로 진술한다(강정찬, 2015; 김재희, 김동호, 2016; 송해덕, 2007).</p>
	<p><b>6. 내용 융합의 원리</b> 융합 요소를 바탕으로 학습 내용, 학습 과제, 성취 기준을 구성한다.</p>
	<p>6.1 교육과정 및 교과서의 내용 요소를 면밀히 검토하고 교과융합 요소를 탐색한다(박영석 외, 2013; 손영수, 이광재, 2016; Weinberg et al., 2001). ex) 관련 교과목의 교과서를 지속적으로 연구하고 연계시킬 수 있는 주제를 찾아 문제해결 학습활동을 설계한다.</p> <p>6.2 컴퓨팅 사고력, 창의력, 문제해결력, 협업능력을 기반으로 통</p>

	<p>합 프로젝트형 주제를 선정한다(강정찬, 2015; 김수환, 2015).</p> <p>6.3 융합 하려는 교과들의 성취기준을 확인 및 분류하여 이를 중심으로 주제를 선정하고 프로젝트를 구성한다(임화영, 윤혜경, 2019; 지경옥, 2017).</p>
<p>학습도구 (창의적 사고도구, 컴퓨팅 도구) 및 정보자원</p>	<p><b>7. 컴퓨팅 도구 지원의 원리</b> 학습자가 흥미롭고 효과적으로 활용할 수 있는 컴퓨팅 도구를 선정한다.</p>
	<p>7.1 초보 학습자도 쉽게 사용하고 산출물을 공유할 수 있는 직관적인 인터페이스의 컴퓨팅 도구(EPL, 파워포인트, 클라우드, 컴퓨터, 태블릿 등)를 선정한다(손영수, 이광재, 2016; 최숙영, 2016; Cross et al., 2013; Quigley &amp; Jamil, 2017; Taljaard, 2016; Weintrop et al., 2016).</p>
	<p>7.2 학습자에게 흥미롭고 동기를 유발 할 수 있는 요소(직관적인 UI, 귀여운 이미지 등)가 포함된 컴퓨팅 도구를 제공한다(Kong, 2016).</p>
	<p>7.3 컴퓨팅 도구를 간단하고 쉬운 수준(초보자용 도구)에서 점진적으로 복잡하고 복합적인 수준(전문가용 도구)으로 제공한다(Cross et al., 2013; Krauss &amp; Prottzman, 2017; Kong, 2016). ex) 프로그래밍을 처음 접하는 학습자들에게는 스크래치 등의 초보자용 도구를 제공하고 점차적으로 학습 수준이 올라가게 되면 파이썬 등의 복잡한 도구를 제공한다.</p>
<p><b>8. 발산·수렴적 사고도구 지원의 원리</b> 창의적 문제해결을 촉진하기 위한 발산·수렴적 사고도구를 지원한다.</p>	
<p>8.1 다양한 아이디어를 구상하고 도출할 수 있도록 발산적 사고도구(브레인스토밍, SCAMPER, 마인드맵 등)를 활용한다(김성일, 2012; 노지예, 이정민, 2018; 전용주, 2017; 박태정, 2018; Firertien, 1982).</p>	

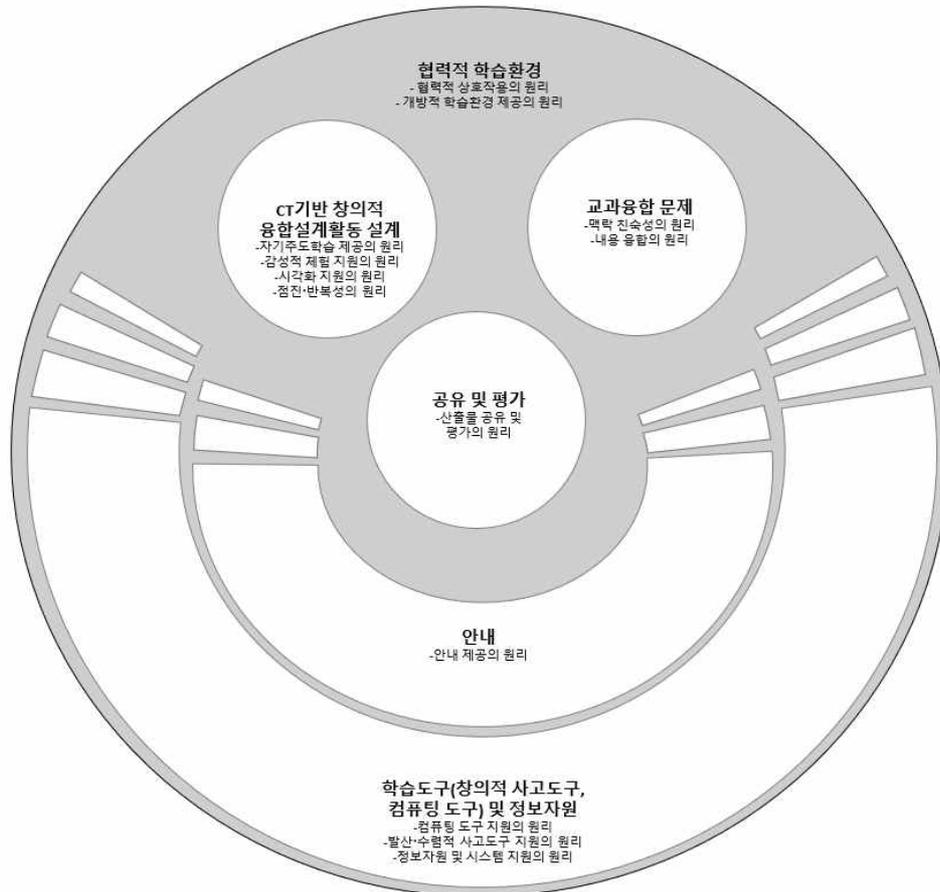
	<p>8.2 문제에 대한 적절한 해결책을 찾을 수 있도록 수렴적 사고도구(HIT, PMI, ALU 등)를 활용한다(김성일, 2012; 노지예, 이정민, 2018; 전용주, 2017; 박태정, 2018; Firetten, 1982).</p>
	<p><b>9. 정보자원 및 시스템 지원의 원리</b>  컴퓨팅 사고력 기반 창의적 융합설계활동을 위한 학습관련 정보 및 온라인 학습지원시스템을 제공한다.</p>
	<p>9.1 학습내용과 학습 활동 지원을 위한 정보자원(프로젝트 관련 내용지식, 코딩명령어 카드 등)을 제공한다(Brennan &amp; Resnick, 2012; Kong, 2016).</p>
	<p>9.2 체계적인 학습자료 및 데이터 접근 방식에 대한 매뉴얼을 지원한다(구양미 외, 2006; 김성기, 1993; 변현정, 2019; 임철일 외, 2014).</p>
	<p>9.3 창의적 문제해결 활동 및 프로젝트 구현을 위한 온라인 학습 공간 및 웹기반 시스템을 제공한다(구양미 외, 2006; 임철일 외, 2009, 2011; 홍미영, 2012; 홍현미, 2017; Chen, &amp; Cheng., 2009; Minamino, &amp; Kinoshita, 2010).</p>
	<p><b>10. 협력적 상호작용의 원리</b>  학습자 간 상호작용을 촉진하기 위한 협력과 동료피드백이 이루어질 수 있도록 지원한다.</p>
협력적 학습환경	<p>10.1 문제해결이 협력적으로 이루어 질 수 있도록 다양한 특성을 가진 구성원의 팀을 구성하여 그룹활동을 촉진한다(임철일 외, 2014; Bacon, Stewart, &amp; Anderson, 2001; Froyd, 2008).</p>
	<p>10.2 학습자들 간 상호 지원활동이 지속적으로 이루어지고 있는지 점검하고 적응적인 피드백을 제공한다(변현정, 2019; Brennan &amp; Resnick, 2012).</p>

	<p><b>11. 개방적 학습환경 제공의 원리</b>          학습자가 자유롭게 사고하고 의견을 공유할 수 있도록 개방적인 학습 분위기를 형성한다.</p> <p>11.1 학습자의 자유로운 사고를 촉진하고 학습자간 의견을 부담 없이 공유할 수 있는 열린 학습 분위기를 형성한다(박태정, 2018; Hamza &amp; Griffith, ,2006; DeSchryver &amp; Yadav, 2015).</p>
공유 및 평가	<p><b>12. 산출물 공유 및 평가의 원리</b>          학습자가 학습활동의 산출물을 제시·공유하여 평가 및 피드백을 받을 수 있는 기회를 제공한다.</p> <p>12.1 창의적 문제해결 과정에 대한 평가기준을 공유한다(변현정, 2019).</p> <p>12.2 프레젠테이션을 통해 산출물을 시연하고 의견을 나눌 수 있도록 안내한다(구덕희, 우석준, 2018; 김재휘, 김동호, 2016; Krauss &amp; Prottzman, 2017).</p> <p>12.3 학습자의 산출물을 다양한 관점에서 객관적으로 평가할 수 있도록 모듈 간·모듈 내 동료평가 및 자기평가 기회를 제공한다 (김재휘, 김동호, 2016; 박태정, 2018; 변현정, 2019; 한국과학창의재단, 2012; Brennan &amp; Resnick, 2012; Lavonen et al., 2004).</p>
안내	<p><b>13. 안내 제공의 원리</b>          학습자가 수업과 관련한 내용을 파악할 수 있도록 수업(학습목표, 운영방식, 프로젝트, 활용 도구, 평가 등)에 대해 안내한다.</p> <p>13.1 학습자가 수업을 하는 과정에서 진행할 내용 및 운영방식을 사전에 안내한다(김혜란, 최선영, 2019; 임철일 외, 2009; Froyd, 2008; Seoane-Pardo, 2016).</p>

	13.2 학습자에게 활용 도구(프로그래밍 언어, 수렴·발산적 사고도구)에 대한 사전 안내를 실시한다(임철일 외, 2014; Firertien, 1982).
	13.3 학습 과제 및 활동에 대해 분명한 준거(과제 수행 시간, 완성도, 평가 계획)를 확립하고 학습자가 항상 열어볼 수 있도록 안내문, 플랫폼 등을 통해 구체적으로 제시한다(변현정, 2019; 손영수, 이광재, 2016; Krauss & Prottzman, 2017).

## 나. 초기 개념모형 개발

선행문헌 고찰을 통해 도출된 구성요소와 설계원리들 간의 개념적 관계를 설명하기 위하여 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계의 초기 개념 모형을 개발하였다([그림 IV-1] 참조). 본 초기 개념모형은 교과-소프트웨어융합 수업설계의 구성요소들 간의 관계를 시각적으로 나타내어 상호관련성을 파악할 수 있도록 하는 데 목적이 있다. 초기 개념모형의 구성요소는 ‘컴퓨팅 사고력(CT)기반 창의적 융합설계활동 설계’, ‘교과융합 문제’, ‘공유 및 평가’, ‘협력적 학습환경’, ‘안내’, ‘학습도구(창의적 사고도구, 컴퓨팅도구) 및 정보자원’ 총 6개로 이루어진다. 개념모형의 구성요소와 구성요소의 관계에 대한 설명은 다음과 같다.



[그림 IV-1] 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 초기 개념모형

‘컴퓨팅사고력(CT) 기반 창의적 융합 설계활동 설계’는 창의적 문제해결을 위한 교과-소프트웨어융합 설계활동을 설계하는 것을 의미한다. 교수자는 학습자가 컴퓨팅 사고력의 핵심 아이디어인 점진적, 반복적 설계를 거둬하며 주도적으로 융합적인 문제와 활동에 도전하고 해결할 수 있도록 수업을 설계해야 한다. ‘교과융합 문제’는 기존 교과와 소프트웨어의 교과 간 융합 수업을 효과적으로 진행하기 위해, 과목을 아우를 수 있는 하나의 맥락과 융합된 교과의 주제를 바탕으로 주어지는 포괄적이고 복합적인 문제이다. ‘협력적 학습 환경’은 창의적 문제해결 과정에서 학습자 간 협력을 통한 상호작용을 촉진하고, 학습자 간 부담 없이 의견

을 주고받을 수 있는 개방적인 학습 환경이다. ‘공유 및 평가’는 학습자가 수업의 산출물을 공유하고 의견을 나누며 피드백을 하고, 이를 평가하는 수업 설계의 구성요소이다. ‘안내’는 효과적인 수업 설계를 위해 수업에 활용하는 사고 도구와 컴퓨팅 시스템 및 수업의 목표, 활동내용, 평가에 대한 전반적인 안내이다. ‘학습도구 및 정보자원’에서 학습 도구는 창의적 문제해결을 위한 교과융합 수업을 진행하기 위해 필요한 사고 도구 및 컴퓨팅 도구, 웹 기반 시스템이며, 정보자원은 컴퓨팅 도구를 활용하기 위한 정보 및 컴퓨팅 도구를 통해 접근 가능한 학습 자료를 의미한다.

‘컴퓨팅 사고력(CT)기반 창의적 융합설계활동 설계’와 ‘교과융합 문제’, ‘공유 및 평가’의 세 요소는 ‘학습도구(창의적 사고도구, 컴퓨팅 도구) 및 정보자원’의 활용과 지원이라는 전제가 필수적이며, 이는 학습도구와 정보자원을 비롯한 수업 전반에 대한 ‘안내’를 포함한다. 이러한 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과융합 수업은 ‘협력적 학습환경’이라는 큰 범주 내에서 이루어져 각 구성요소에 영향을 미친다.

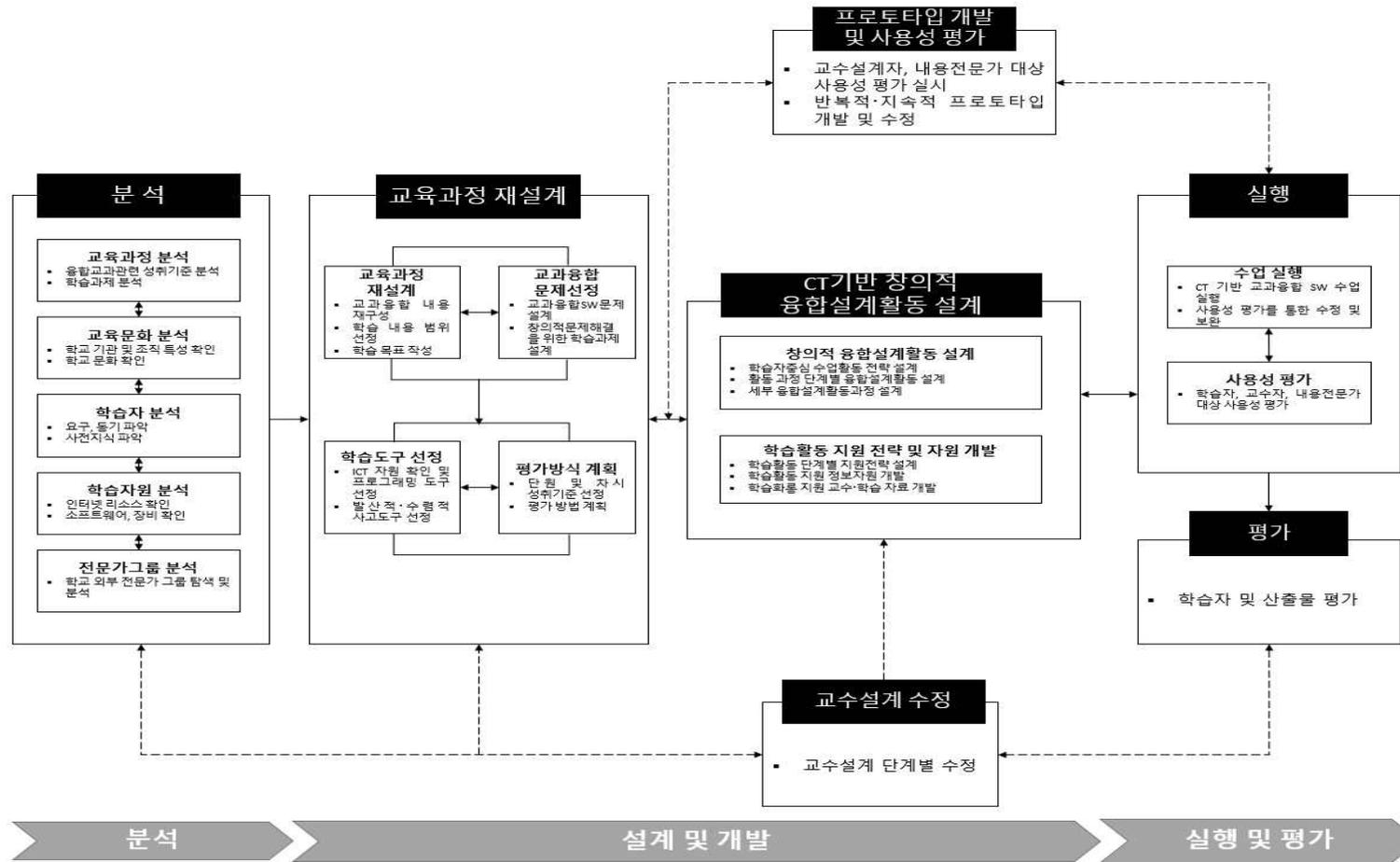
#### 다. 초기 절차모형 개발

본 연구에서 절차모형은 교수자가 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과융합 수업을 설계할 때 거쳐야 하는 주요 단계와 각 단계마다 고려해야 할 활동은 무엇인지를 안내함으로써 수업 설계의 전체적인 과정과 절차를 안내하는 것을 목적으로 한다. 초기 절차 모형은 본 연구와 관련 있는 설계모형이 지닌 주요 특성과 시사점을 기반으로 ADDIE 모형의 ‘분석’, ‘설계 및 개발’, ‘실행 및 평가’의 각 단계와의 관련성을 검토하여 개발되었으며, 각 단계마다 세부적인 설계활동이 포함되었다. 주요 설계모형의 특성 및 시사점을 절차별로 유목화 하여 종합한 결과는 <표 IV-3>와 같다.

<표 IV-3> 주요 설계모형의 특성 및 시사점의 절차별 유목화

주요 내용	분석	설계	개발	실행	평가
학습자 및 학습환경 분석(이소이, 노태천, 2011; 손영수, 이광재, 2016; Smaldino et al., 2008)	●				
실생활 맥락의 문제 선정(함성진, 김순화, 송기상, 소효정, 2017; Stohlmann et al., 2012)		●			
교육과정 및 학습내용 재설계(김진수, 2012; 이소이, 노태천, 2011; Weinberg et al., 2001)		●			
활용이 쉬운 컴퓨팅 도구 및 사고 도구 선정(구양미 외, 2006; Kong, 2016; Wang & Woo, 2007)		●			
다양한 정보자원 제공(구양미 외, 2006; Kong, 2016)		●			
문제 정의, 분해 및 아이디어 생성 지원(전용주, 2017; 함성진 외, 2014; Isaksen, Dorval, & Treffinger, 2000; Wells, 2015)			●		
아이디어 조직 및 시각화 지원(전용주, 2017)			●		
문제 분석 및 분해, 자동화와 추상화 지원(김진숙 외, 2015; 전수진, 2017; Curzon et al., 2014; Kong, 2016)			●		
점진적·반복적 문제 해결 지원(Brennan & Resnick, 2012; Kong, 2016)			●		
프로토타입 개발 및 사용성 평가(배선아, 2011; 임철일, 연은경, 2006)				●	
학습자 참여(구재훈, 김태영, 2018; 손영수, 이광재, 2016; 임철일, 홍미영, 이선희, 2011)				●	
산출물 공유 및 공유 플랫폼 제공(임철일 외, 2011; 함성진, 김순화, 송기상, 2014)					●
다각적 평가 진행(구양미 외, 2006; 배학진 외, 2009; 이동희 외, 2015; Brennan & Resnick, 2012; Robertson, 2013)					●

또한 컴퓨팅 도구를 활용하여 다양한 활동들을 하게 되며, 소프트웨어와 기존 교과 간의 융합 수업이라는 익숙지 않은 새로운 형태의 수업을 수행해야 한다는 측면에서, 실제 수업을 실행하기에 수업 설계와 관련한 동료 교사 및 전문가, 학습자의 의견을 듣고 반영할 필요가 있다. 따라서 래피드 프로토타입(Rapid Prototyp) 개발 방법론(임철일 연은경, 2006)을 절차모형에 적용하여 지속적으로 개선된 산출물을 개발할 수 있도록 하였다.([그림 IV-2])



[그림 IV-2] 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업설계 초기 절차모형

절차모형의 분석 단계는 교육과정 분석, 교육문화 분석, 학습자 분석, 학습자원 분석, 전문가그룹 분석으로 이루어진다. 먼저 교육과정 분석 단계에서는 소프트웨어와 융합을 할 수 있는 융합교과와 관련한 교육과정의 성취기준에 따른 학습 요소와 과제를 분석한다. 교육문화 분석 단계에서는 정규 수업 시간에 학습자가 개인 컴퓨터를 활용하여 교과-소프트웨어융합 수업에 대한 학교, 기관, 조직 및 학부모의 허용 상황과 선호도, 지원 정도를 확인한다. 학습자 분석 단계에서는 컴퓨팅 사고력에 기반한 소프트웨어 수업, 교과융합 수업에 대한 학습자의 요구와 흥미, 동기를 파악하고 소프트웨어 수업과 관련한 학습 경험 및 사전 지식을 확인한다. 학습자원 분석 단계에서는 학교에 설치된 인터넷 리소스를 확인하고, 개인 PC 혹은 노트북, 태블릿 PC 등의 장비와 EPL 소프트웨어, 학습지원플랫폼 등의 설치 여부를 확인한다. 전문가그룹 분석 단계에서는 교과융합 소프트웨어 수업 설계에 대해 지식과 조언을 구하고 사용성 평가에 참여할 수 있는 수업과 관련한 학교 내외부의 전문가 그룹을 탐색하고 영입할 수 있는 전문가를 파악한다.

이 후 설계 및 개발 단계는 교육과정 재설계와 창의적 융합활동 설계 단계로 구성된다. 교육과정 재설계 단계는 두 개의 하위 단계로 이루어진다. 우선 첫 번째 하위 단계는 융합할 과목과 단원의 학습 내용의 범위를 선정하고, 범위 안의 내용을 융합 및 통합하여 차시의 구성 및 순서를 조정하는 교육과정 재설계 단계와, 이에 따라 수업에서 다룰 핵심적인 교과융합 소프트웨어 문제를 선정하는 교과융합 문제선정 단계로 구성된다. 이후 두 번째 하위 단계는 실제 수업에서 사용하는 컴퓨터, 태블릿 PC 등의 하드웨어와 엔트리, 스크래치 등의 EPL 소프트웨어를 선정하고, 교과융합 문제 해결을 위해 학습자들이 활용 가능한 발산적, 수렴적 사고 도구를 선정하는 학습도구 선정 단계와, 단원 및 성취기준에 따라 평가방법을 계획하는 평가방식 계획 단계로 구성된다. 이후 본격적인 융합활동 설계 단계를 거친다. 컴퓨팅 사고력 기반 창의적 융합활동 설계 단계에서는 학습자들이 수행할 세부적인 단계별 융합 설계활동을 설계하고, 학습지원 전략과 정보 자원, 교수학습 자료를 개발한다.

실행 및 평가 단계에서는 실제 교과-소프트웨어융합 수업을 실행하고, 학습자 및 산출물을 평가한다. 이때 수업 실행 이후 학습자, 교수자 및 내용전문가 대상의 사용성 평가를 통해 개선점을 도출한 후, 수정 및 보완 작업을 거친다. 특히 지속적, 반복적인 프로토타입 개발과 사용성 평가를 통해 설계, 개발하여 최적화된 수업을 설계하는 프로토타입 및 사용성 평가 단계가 창의적 융합설계활동 설계 단계부터 수업 실행까지 순환적으로 이루어진다. 수업 이후 도출된 수정 및 개선점은 교수설계 수정 단계를 거쳐 다음 수업 설계에 반영한다.

## 2. 내적 타당화

도출된 초기 구성요소와 설계원리, 개념모형, 절차모형의 내적 타당성을 확인하기 위해 총 두 차례에 걸쳐 전문가 타당화와 한 차례의 사용성 평가가 이루어졌다. 내적 타당화를 진행한 과정과 결과물을 요약하면 다음 <표 IV-4>와 같다.

<표 IV-4> 내적 타당화를 위한 주요 내용 요약

구분	대상	참여 인원	검토내용	결과물
1차 전문가 타당화	교수설계 전문가	3	<ul style="list-style-type: none"> <li>초기 모형 도출을 위한 선행문헌 탐색과 해석의 적절성에 대한 타당화</li> <li>구성요소에 대한 타당화</li> <li>설계원리 전반에 대한 타당화</li> </ul>	<ul style="list-style-type: none"> <li>초기모형 수정사항 도출</li> <li>2차 설계원리 및 모형 개발</li> </ul>
	소프트웨어 교육 전문가	2	<ul style="list-style-type: none"> <li>개별 설계원리 및 지침에 대한 타당화</li> <li>개념 모형에 대한 타당화</li> <li>절차 모형에 대한 타당화</li> </ul>	
2차 전문가 타당화	교수설계 전문가	3	<ul style="list-style-type: none"> <li>구성요소에 대한 타당화</li> <li>설계원리 전반에 대한 타당화</li> </ul>	<ul style="list-style-type: none"> <li>2차 모형 수정사항 도출</li> <li>3차 설계원리 및 모형 개발</li> </ul>
	소프트웨어 교육 전문가	2	<ul style="list-style-type: none"> <li>개별 설계원리 및 지침에 대한 타당화</li> <li>개념 모형에 대한 타당화</li> <li>절차 모형에 대한 타당화</li> </ul>	
사용성 평가	초등학교 현장교사	3	<ul style="list-style-type: none"> <li>연구결과물이 설계에 도움이 되는지 여부</li> </ul>	<ul style="list-style-type: none"> <li>3차 모형 수정사항 도출</li> </ul>

- 연구결과물의 강점, 약점, 개선점
- 개별 설계원리에 대한 개발에 반영 평가
- 최종 모형

## 가. 구성요소에 관한 전문가 타당화

### 1) 1차 전문가 타당화 결과

초기 모형의 구성요소에 대한 1차 전문가 타당화를 위해서 두 부류의 전문가 집단 즉, 교수설계 전문가 3인과 컴퓨터 교육 및 소프트웨어 교육 전문가 2인이 참여하였다. 교수설계 전문가 3인은 모두 교육학 박사 소지자로서 교수설계분야의 경력이 7년 이상이며, 컴퓨터 교육 및 소프트웨어 교육 전문가 2인은 컴퓨터 교육 석사 및 박사 학위 소지자로 현재 소프트웨어 교육 전문가로 10년 이상의 경력을 지닌 전문가이다.

<표 IV-5> 구성요소에 대한 1차 전문가 타당화 결과

영역	전문가					평균	CVI	IRA
	A	B	C	D	E			
선택문헌 탐색의 적절성	2	2	3	2	2	2.2	0.20	
선행문헌 고찰 결과 요약 및 해석의 적절성	3	2	4	2	2	2.6	0.40	
- 구성요소의 적절성								
· 구성요소의 타당성	2	2	3	3	2	2.4	0.40	
· 구성요소의 수준	2	2	2	3	2	2.2	0.20	
· 하위요소의 구성	3	2	3	3	3	2.8	0.80	0.00
- 구성요소별 설명성								
· 컴퓨팅사고력 기반 창의적 융합설계활동 설계	2	2	3	2	2	2.2	0.20	
· 교과융합 문제	2	2	4	3	2	2.6	0.40	
· 학습도구(창의적 사고도구, 컴퓨팅도구) 및	2	2	3	2	3	2.4	0.40	

정보자원							
· 공유 및 평가	2	2	3	3	3	2.6	0.60
· 협력적 학습환경	2	2	3	2	3	2.4	0.40
· 안내	2	2	2	2	3	2.2	0.20

구성요소 도출에 대한 1차 전문가 타당화 결과 11개의 평가 문항의 평균이 모두 평균 3.0 이하의 낮은 수준으로 나타났다. 이는 초기 도출한 구성요소의 정의와 범주, 수행 주체에 대한 설명이 명확하지 않고, 구성요소와 하위원리 간의 위계가 불분명하여 유목화 및 구조화가 필요하고 전체적으로 일반적인 테크놀로지 활용 수업의 설계원리와 차별점이 없다는 전문가들의 판단이 반영된 것으로 해석할 수 있다. 또한 각 평가문항에 대한 내용타당도 지수(Content Validity Index : CVI)를 살펴보면, 11개의 평가 문항 모두 0.80 이하인 것으로 나타났다. 일반적으로 CVI 값이 .80 이상이면 타당하다고 해석하나, 타당화 결과 모두 .80 이하의 점수를 받아 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과융합 소프트웨어 수업 설계를 위해 도출된 구성요소는 타당성이 낮다고 판단할 수 있다. 한편, 평정자간 일치도 지수(Inter-Rater Agreement : IRA)는 전문가들의 평가에 대한 신뢰도를 결정하는 지수로 .80이상이면 전문가들의 평가를 신뢰할 수 있다고 판단할 수 있다(홍미영, 2012). 도출된 구성요소에 대한 IRA 지수는 0.00점인 것으로 나타나 전문가들의 평가에 대한 신뢰도가 낮은 것으로 판단할 수 있다.

구성요소에 관한 전문가 타당화를 통해 수합된 수정 및 보완 사항을 정리하면 다음과 같다. 첫째, 구성요소들의 위계가 맞지 않으므로 개별 구성요소와 하위 설계원리를 재구조화 및 유목화 하여 포괄적인 상위 구성요소를 다시 도출해야 할 필요성이 있다. 예컨대 구성요소 중 ‘안내’의 경우, 상위 개념의 구성요소보다 각 요소의 하위 원리에 더 적합하다는 전문가의 검토의견이 있었다. 둘째, 구성요소에 대한 정의와 설명이 명확히 제시되어야 한다. 즉, 컴퓨팅 사고력, 창의적 문제해결, 교과융합 문제와 같은 요소에 대한 개념과 설명을 명확히 제시하고, 중심을 두어야 하는 요소를 도출하기 위해 정의를 제한 할 필요하다는 전문가의 검토의견

이 있었다. 셋째, 각 구성요소를 이루는 하위요소를 추가하고 이에 대한 명확한 내용을 추가할 필요성이 있다. 도출된 초기 구성요소는 다소 넓은 범위의 개념을 포괄하고 있으며 각 구성요소에 대한 내용이 뚜렷이 드러나지 않으므로, 주요 하위 요소를 제시하여 핵심적으로 고려해야 할 사항을 명확히 전달할 수 있도록 수정해야 한다. 마지막으로 구성요소를 실행하는 주체가 명확하지 않으므로, 표현을 수정해야 할 필요가 있다. 융합 설계활동 설계와 공유 및 평가, 안내와 같은 구성요소가 교수자 혹은 학습자 중 어느 주체가 실행하는 것인지를 고려하여 구성요소에 대한 표현과 설명을 수정해야 한다는 의견이 제기되었다. 이상의 전문가 검토 의견을 반영하여 초기 모형의 구성요소를 <표 IV-6>과 같이 수정하였다.

<표 IV-6> 1차 전문가 타당화를 통해 수정된 2차 모형의 구성요소

초기 모형의 구성요소	2차 모형의 구성요소 및 하위요소	
컴퓨팅 사고력 기반 창의적 융합 설계활동 설계	창의적 교과-소프트웨어 융합 설계활동 지원	<ul style="list-style-type: none"> <li>• 문제이해 지원</li> <li>• 아이디어 생성 지원</li> <li>• 논리화 지원</li> <li>• 문제해결 지원</li> </ul>
교과융합 문제	교과융합 교육과정	<ul style="list-style-type: none"> <li>• 성취기준</li> <li>• 교과공통주제</li> <li>• 융합과제</li> <li>• 수업구성</li> </ul>
학습도구 및 정보자원	학습도구 및 정보자원	<ul style="list-style-type: none"> <li>• 소프트웨어</li> <li>• 하드웨어 자원</li> <li>• 수업 안내 자료</li> <li>• 수렴발산 지원 도구</li> <li>• 디지털 정보자료</li> <li>• 이러닝 플랫폼</li> </ul>

공유 및 평가	공유 및 평가 지원	<ul style="list-style-type: none"> <li>• 공유 방법</li> <li>• 평가 기준</li> <li>• 학습자 성찰</li> </ul>
협력적 학습 환경	삭제	
안내	삭제	

## 2) 2차 전문가 타당화 결과

1차 전문가 타당화를 통해 2차 모형의 구성요소를 도출하였다. 수정된 구성요소에 대한 2차 전문가 검토 결과 모든 항목에서 평균 3.4점 이상인 것으로 나타나 창의적 문제해결을 위한 교과-소프트웨어 융합 수업 설계를 위해 도출된 구성요소는 타당한 것으로 판단할 수 있다. CVI는 각 설문 항목에 대해 타당하다고 판단하는 전문가의 비율을 제공해 주는데, 아홉 개의 항목 모두 1.00으로 전문가의 평가가 일치하는 것으로 나타났다. 전문가들의 평가를 얼마나 신뢰할 수 있는지 결정할 수 있는 지수인 IRA도 1.00으로 나타나 2차 전문가 타당화의 결과가 신뢰할 수 있는 것으로 나타났다.

<표 IV-7> 구성요소에 대한 2차 전문가 타당화 결과

영역	전문가					평균	표준 편차	CVI	IRA
	A	B	C	D	E				
선택문헌 탐색의 적절성	4	4	4	3	4	3.80	.447	1.00	
선행문헌 고찰 결과 요약 및 해석의 적절성	4	4	4	3	4	3.80	.447	1.00	
- 구성요소의 적절성									1.00
· 구성요소의 타당성	4	3	4	3	3	3.40	.548	1.00	
· 구성요소의 수준	4	3	4	3	4	3.60	.548	1.00	
· 하위요소의 구성	4	3	4	3	4	3.60	.548	1.00	

- 구성요소 별 설명성									
· 교과융합 교육과정	4	3	4	3	4	3.60	.548	1.00	
· 학습도구 및 정보자원	4	3	3	3	4	3.40	.548	1.00	
· 창의적 교과-소프트웨어융합 설계활동 지원	4	3	4	3	4	3.60	.548	1.00	
· 공유 및 평가 지원	4	3	4	3	3	3.40	.548	1.00	

개선이나 보완에 관한 연구자의 질문에 대해 전문가는 본 연구가 교과-소프트웨어에 대한 내용임을 분명히 드러나도록 구성요소에 해당 표현을 포함시킬 필요가 있으며, 일부 개별 원리의 내용은 상위 구성요소의 성격을 띠고 있으므로 위계를 변경하여 구성요소를 추가하는 것을 제안하였다. 따라서 구성요소의 큰 틀은 유지하되 일부 표현을 수정하고, 기존 개별 원리와 상세지침을 재구조화하여 새로운 구성요소로 변경하는 것으로 3차 모형의 구성요소를 도출하였다.

<표 IV-8> 2차 전문가 타당화를 통해 수정된 3차 모형의 구성요소

2차 모형의 구성요소 및 하위요소		3차 모형의 구성요소 및 하위요소	
교과융합 교육과정	<ul style="list-style-type: none"> <li>성취기준</li> <li>교과공통주제</li> <li>융합과제</li> <li>수업구성</li> </ul>	교과-소프트웨어 융합 교육과정	<ul style="list-style-type: none"> <li>성취기준</li> <li>교과공통주제</li> <li>교과내용</li> <li>교육과정 재조직</li> </ul>
학습도구 및 정보자원	<ul style="list-style-type: none"> <li>소프트웨어</li> <li>하드웨어 자원</li> <li>수업 안내 자료</li> <li>수렴발산 지원 도구</li> <li>디지털 정보자료</li> </ul>	교과-소프트웨어 융합 학습도구	<ul style="list-style-type: none"> <li>프로그래밍 소프트웨어 자원</li> <li>하드웨어 자원</li> <li>수렴/발산 사고 지원 도구</li> <li>공동작업 및 토론 지원 이터닝</li> </ul>

	• 이러닝 플랫폼		플랫폼
-		교과-소프트웨어 융합 학습지원	<ul style="list-style-type: none"> <li>• 학습방법 안내</li> <li>• 모듈구성</li> <li>• 학습자료</li> <li>• 수업규칙</li> </ul>
창의적 교과-소프트웨어 융합 설계활동 지원	<ul style="list-style-type: none"> <li>• 문제이해 지원</li> <li>• 아이디어 생성 지원</li> <li>• 논리화 지원</li> <li>• 문제해결 지원</li> </ul>	교과-소프트웨어 융합 수업활동	<ul style="list-style-type: none"> <li>• 문제이해 지원</li> <li>• 아이디어 생성 지원</li> <li>• 논리화 지원</li> <li>• 문제해결 지원</li> </ul>
공유 및 평가 지원	<ul style="list-style-type: none"> <li>• 공유 방법</li> <li>• 평가 기준</li> <li>• 학습자 성찰</li> </ul>	공유 및 평가	<ul style="list-style-type: none"> <li>• 시연 방식 선정</li> <li>• 공유 방법 선정</li> <li>• 평가기준 선정</li> </ul>

## 나. 설계원리에 관한 전문가 타당화

### 1) 설계원리 전반에 대한 전문가 타당화 결과

설계원리 전반에 대한 전문가 타당화는 원리의 타당성, 설명력, 유용성, 보편성, 이해도로 구성된 다섯 가지 항목을 검토하고 이에 대한 의견을 묻는 방법으로 진행되었다. 설계원리 전반에 대한 전문가 타당화 결과는 아래 <표 IV-9>와 같다.

<표 IV-9> 설계원리 전반에 대한 전문가 타당화 결과

	1차 전문가 타당화 (사례수=5)				2차 전문가 타당화 (사례수=5)			
	평균	표준 편차	CVI	IRA	평균	표준 편차	CVI	IRA
타당성	2.40	.548	0.40		3.80	.447	1.00	
설명력	2.00	.000	0.60		3.60	.548	1.00	
유용성	3.20	.447	0.20	0.00	3.60	.548	1.00	1.00
보편성	1.20	.447	0.40		3.60	.548	1.00	
이해도	2.60	.548	0.20		3.80	.447	1.00	

설계원리 전반에 대한 1차 전문가 타당화 검토 결과 유용성이 3.2점으로 가장 높게 나타났으나, 그 외의 항목은 1.20점에서 2.40 사이에 분포하여 낮은 수준의 타당도를 보였다. CVI는 다섯 항목 모두 0.2에서 0.6 사이로 낮게 나타났으며, 특히 개별 전문가 중 두 명이 타당성, 설명력, 유용성, 보편성, 이해도 측면에서 전반적으로 부정적인 평가점수를 제시하여 전체적인 설계원리에 대한 수정이 필요한 것으로 해석할 수 있다. 전반적인 설계원리에 대한 전문가의 검토 의견은 첫째, 도출한 원리를 토대로 무리 없이 수업을 설계할 수 있도록 구체적인 설명과 예시자료 추가가 필요하다는 것이다. 둘째는 연구의 핵심을 드러내는 설계원리가 필요하다는 것이다.

2차 전문가 타당화에서는 예시 및 해설이 추가되었고, 창의적 문제해결을 위한 교과-소프트웨어융합 수업에 대한 독특성을 드러내는 설계원리 및 상세지침을 추가적으로 삽입하였으며 이에 대한 검토가 이루어졌다. 설계원리에 전반에 대한 전문가 타당화 결과 타당성(평균 3.80점), 설명력(평균 3.60점), 유용성(평균 3.60점), 보편성(평균 3.60점), 이해도(평균 3.80점)으로 모든 항목에서 높은 점수를 보였다. CVI는 모두 1.00으로 각 항목에 대해 타당화에 참여한 모든 전문가가 타당하다고 응답하였으며, IRA도 1.00으로 나타나 전문가들의 평가가 일관성을 가지고 있으며, 평가 결과를 신뢰할 수 있는 것으로 판단된다.

## 2) 개별 설계원리에 대한 전문가 타당화 결과

### 가) 1차 전문가 타당화 결과

총 13개의 설계원리와 32개의 상세 지침으로 구성된 초기 설계원리에 대한 1차 전문가 타당화 결과는 <표 IV-10>와 같으며, 개별 상세 지침에 대한 타당화 결과는 [부록 3]에 제시하였다.

<표 IV-10> 초기 개별 설계원리에 대한 1차 전문가 타당화 결과

구성요소	설계원리	전문가					평균	CVI	IRA
		A	B	C	D	E			
CT활용 창의적 융합설계 활동	1. 자기주도학습 체 공의 원리	2	2	3	1	3	2.20	0.40	0.00
	2. 감성적 체험 지원 의 원리	1	2	2	1	3	1.80	0.20	0.00
	3. 시각화 지원의 원 리	2	2	4	1	3	2.40	0.40	0.00
	4. 점진·반복성의 원 리	2	2	2	1	3	2.00	0.20	0.00
교과융합 문제	5. 맥락 친숙성의 원 리	2	2	3	1	3	2.20	0.40	0.00
	6. 내용 융합의 원리	1	2	2	2	3	2.00	0.20	0.00
학습도구( 창의적 사고도구, 컴퓨팅도 구) 및 정보자원	7. 컴퓨팅 도구 지원 의 원리	2	2	2	1	3	2.00	0.20	0.00
	8. 발산/수렴적 사고 도구 지원의 원리	1	2	3	1	3	2.00	0.40	0.00
	9. 정보자원 및 시스 템 지원의 원리	2	2	2	2	3	2.20	0.20	0.00
개방적 학습환경	10. 협력적 상호작용 의 원리	1	2	3	1	3	2.00	0.40	0.00
	11. 개방적 학습환경 제공의 원리	3	2	2	1	3	2.20	0.40	0.00
공유 및 평가	12. 산출물 공유 및 평가의 원리	2	2	3	1	3	2.20	0.40	0.00
안내	13. 안내 제공의 원 리	3	2	2	1	3	2.20	0.40	0.00

총 13개의 개별 설계원리에 대한 1차 전문가 타당화 검토 결과, 평균 점수는 1.80점~2.40점 사이에 분포하고 있는 것으로 나타난다. 또한 총

32개의 상세지침에 대한 평균점수는 1.80점~2.60점 사이에 분포하고 있는 것으로 나타난다. 따라서 초기 개별원리와 상세지침은 전체적으로 수정할 필요성이 있는 것으로 해석할 수 있다. 개별 설계원리에 대한 CVI는 0.2~0.4 사이에 분포하고 있어 개별 설계원리에 대한 전문가들의 평가가 일치하지 않는 것으로 나타났으며, 상세지침의 경우도 0.2~0.6 사이에 분포하고 있고, IRA도 모두 0.00으로 나타나 전체적으로 수정과 보완, 삭제가 필요함을 알 수 있다. 1차 전문가 타당화 검토 의견과 수정 및 개선사항에 대한 내용을 정리하면 다음 <표 IV-11>와 같다.

<표 IV-11> 1차 전문가 타당화 검토 의견과 수정 및 개선사항

전문가 의견 및 수정점		세부 개선사항
재구조화	구성요소 재구조화	<ul style="list-style-type: none"> <li>▪ 교과융합 교육과정, 학습도구 및 정보자원, 창의적 교과-융합설계 활동지원, 공유 및 평가로 구성요소 수정</li> </ul>
	원리 및 상세지침 추가 및 삭제	<ul style="list-style-type: none"> <li>▪ 원리 1, 2, 3, 4, 6, 7, 8, 10 추가</li> <li>- 상세지침 1.1, 1.2, 4.2, 5.3, 6.1, 6.2, 6.3, 8.1, 8.3, 9.2, 10.2, 10.3 추가</li> <li>▪ 원리 1, 2, 3, 5, 10, 11, 13 삭제</li> <li>- 상세지침 1.1, 1.2, 2.1, 2.2, 2.3, 3.2, 6.2, 7.2, 10.2, 13.2 삭제</li> </ul>
	원리 및 상세지침 위치 조정	<ul style="list-style-type: none"> <li>▪ 1차 원리 4를 2차 원리 9로 위치 조정</li> <li>▪ 1차 원리 12를 2차 원리 11로 위치 조정</li> <li>- 4.1 → 9.1, 5.2 → 2.3, 7.1 → 4.1, 8.1 → 7.1, 8.2 → 7.2, 11.1 → 11.5,</li> </ul>
	원리 및 상세지침 통합 및 세분화	<ul style="list-style-type: none"> <li>▪ 1차 원리 6을 2차 원리 3으로 구체화</li> <li>▪ 1차 원리 7, 9를 2차 원리 5로 통합</li> <li>- 4.2 → 7.1, 7.2 세분화</li> <li>- 5.1, 5.2 → 2.2, 3.2 통합</li> <li>- 6.1 → 2.1, 3.1 세분화</li> <li>- 9.1 → 5.2, 5.4, 8.2 세분화</li> <li>- 12.2 → 11.1, 11.2 세분화</li> </ul>

		- 12.3 → 11.3, 11.4 세분화 - 13.1, 13.3 → 5.4 통합
진술형식 수정	내용 진술 구체화	<ul style="list-style-type: none"> <li>▪ 내용 진술 구체화</li> <li>- 6.3 → 2.1, 5.4의 내용 진술 구체화</li> <li>- 9.2 → 5.4의 내용 진술 구체화</li> <li>- 9.3 → 5.1의 내용 진술 구체화</li> <li>- 12.1 → 5.4의 내용 진술 구체화</li> </ul>
	예시 추가	<ul style="list-style-type: none"> <li>▪ 모든 세부지침에 예시 추가</li> </ul>

초기 원리 및 상세지침은 1차 전문가 타당화 의견을 반영하여 전체적으로 재구성, 수정, 통합, 세분화, 삭제되었으며, 총 11개의 원리와 32개의 세부지침으로 구성된 2차 설계원리 및 상세지침을 도출하였다. 2차 설계원리 및 상세지침 도출 시 반영된 1차 전문가 타당화 의견과 개선사항을 정리하면 다음과 같다.

첫째, 1차 전문가 타당화 결과 전체적인 수업설계 원리 및 상세지침이 창의적 문제해결 모형(CPS모형)의 단계를 반영하여 개선될 필요가 있다는 의견을 확인하였다. 이에 따라 Isaksen, Dorval, & Treffinger(2000)의 CPS 모형, 전용주(2018)의 CT-CPS 수업모형과 함성진, 김순화, 송기상(2014)의 CT-STEAM 모형 등을 비교하여 내용적으로 유사하거나 관련있는 단계를 포괄할 수 있는 핵심 요소를 도출하였으며, 이를 2차 수업설계원리의 개선에 적용하였다. 자세한 도출 과정은 [부록 5]에 포함하였다. 둘째, 1차 전문가 타당화 의견에 따라 본 연구의 핵심을 드러낼 수 있는 원리와 상세지침을 추가하고, 타당화 점수가 낮게 나타나거나 수정이 필요하다는 전문가 의견이 있는 부분은 이를 반영하여 삭제하였다. 원리와 상세지침의 추가에 따라 기존 초기 내용과 중복되거나 유사한 경우는 삭제하였다. 셋째, 구성요소의 변경과 원리 및 상세지침의 추가, 삭제에 따라 전체적으로 위치를 조정하고 재구성하였다. 넷째, 기존의 초기 원리와 상세지침의 내용을 통합하거나 세분화 하였다. 하나의 원리와 상세지침에 내용이 너무 적거나 많다는 전문가의 의견이 있는 경우 이를

반영하여 통합 및 세분화하였다. 다섯째, 내용의 진술 형식을 구체화하였다. 설계원리는 상세지침의 내용을 포괄하는 상위 개념으로서 내용의 진술을 보다 명확하게 수정하였으며, 상세지침 중 모호하고 추상적으로 서술된 부분은 구체적인 행위를 나타내는 동사로 수정하였다. 마지막으로 실제 수업 설계 시 구체적인 가이드라인을 제공하기 위해 모든 세부 지침에 예시를 추가하여 이해를 돕고자 하였다. 이와 같은 사항을 반영하여 수정된 2차 설계원리 및 상세지침은 <표 IV-12>와 같다.

<표 IV-12> 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계원리와 상세지침 - 2차

구성요소	원리 및 상세지침	
교과융합 교육과정	<b>1. 적절성 확인의 원리</b> 소프트웨어와 연계할 교과/단원의 학습내용 및 성취기준을 확인하고 교과융합소프트웨어 수업의 효과성 및 실현가능성을 판단한다.	
	1.1 융합하려는 교과 및 단원의 성취기준, 학습내용, 교수학습 방법, 평가방법, 학습자 수준을 확인하고 교과융합소프트웨어 수업이 교과융합 소프트웨어 수업으로 구현할 수 있는지를 판단한다.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 기존 과목과 소프트웨어수업 내용을 융합하는 것이 도움이 되는가?</li> <li>• 학습자가 교과융합 소프트웨어 수업 내용을 이해하고 산출물을 만들어 낼 수 있는가?</li> <li>• 교과융합 소프트웨어교육을 통해 기대되는 효과는 무엇인가?</li> </ul>
	1.2 학습 환경, 자원, 학교 방침 및 분위기, 전문가 그룹 등 수업을 위한 제반 자원과 환경을 파악하고 교과융합 소프트웨어 수업의 실행 및 지원 가능성을 확인한다.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 학교 방침과 문화가 교과융합 소프트웨어 수업을 실행하기에 적절한가?</li> <li>• 교과융합 소프트웨어 실행을 위해 동료 교사의 이해와 협조가 적절히 이루어지는가?</li> <li>• 교과융합 소프트웨어 수업 실행을 위해 시간과</li> </ul>

	<p>자원, 비용이 얼마나 소요되는가?</p> <ul style="list-style-type: none"> <li>교과융합 소프트웨어 수업을 설계하고 실행하기 위해 조언을 받을 수 있는 전문가 그룹(동료 교사, 프로그래밍 전문가)이 있는가?</li> </ul>
<p><b>2. 융합 문제 설계의 원리</b></p> <p>융합 과목의 소프트웨어 연계요소를 고려하여 학습자가 이해하기 쉽고 문제 해결활동에 몰입할 수 있는 교과융합 문제를 설계한다.</p>	
<p>2.1 융합하려는 교과목의 성취기준, 학습내용, 소프트웨어 수업 요소와의 연계 요소와 학습자 수준을 고려하여 공통주제, 수업 목표, 과제 형식을 선정한다.</p>	
<p>예시 및 해설</p>	<p>○ [6사05-03] 초등학교 6학년 사회과목의 ‘우리나라의 정치 발전’ 단원 중 ‘일상생활과 민주주의’</p> <ul style="list-style-type: none"> <li>성취기준 : 일상생활에서 경험하는 민주주의 실천 사례를 탐구하여 민주주의의 의미와 중요성을 파악하고, 생활 속에서 민주주의를 실천하는 태도를 기른다.</li> <li>공통학습주제 : 민주주의 실천을 위한 전자 투표 프로그램 만들어 보기</li> <li>소프트웨어수업 연계 학습 목표 : 1) 전자투표 프로그램을 설계할 수 있다. 2) 리스트와 조건문을 활용하여 프로그램을 만들 수 있다. 3) 만든 전자 투표 프로그램을 활용하여 투표를 실시할 수 있다.</li> </ul> <p>(엔트리 학습자료 ‘전자투표 프로그램으로 모의선거 실시하기’ 자료 참고)</p>
<p>2.2 실생활의 상황을 바탕으로 학습자가 친숙함을 느낄 수 있는 시나리오 형태로 문제를 제시한다.</p>	
<p>예시 및 해설</p>	<ul style="list-style-type: none"> <li>[6실03-04] 초등학교 6학년 실과 과목의 ‘자원 관리와 자립’ 단원 내용을 중심으로 환경오염으로 발생하는 생활 주변의 문제점들을 알아보고, 쓰레기 재활용을 통한 환경 보호의 필요성을 안내 후, 분리수거와 관련된 문제를 해결할 수 있는 프로그램 설계하기</li> </ul> <p>(엔트리 학습자료 ‘자동으로 분리수거를 해요’ 자료</p>

	참고)
	2.3 해결안이 하나 이상으로 제시될 수 있는 복합적이고 비구조화된 형태의 문제를 설계한다.
예시 및 해설	<ul style="list-style-type: none"> <li>비구조화 형태의 문제 설계 시 고려할 점 : 1) 문제와 관련된 상황이나 요소가 분명히 정의되어 있지 않은 문제로 설계한다. 2) 문제 해결에 필요한 정보가 충분히 포함되어 있지 않은 문제를 설계한다. 3) 다양한 해답이나 해결 경로를 가지고 있다. 4) 의사결정의 명료화와 논쟁에 초점을 둔다.</li> </ul>
<b>3. 수업 재구성의 원리</b>	
융합할 교과/단원의 성취기준을 바탕으로 교과융합소프트웨어 수업의 주제를 선정하고 수업을 재구성한다.	
3.1 교과 특성, 주제의 연계성, 학습자 수준을 고려하여 단원의 순서를 바꾸거나 내용을 생략하고 주제와 관련된 주요 내용을 중심으로 수업을 재구성한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>수업 재구성 : 각 교과의 성취기준을 확인하고 공통 주제를 선정 후, 통합된 학습 내용을 중심으로 교육과정 상의 수업 순서를 바꾸거나 주요 내용을 다루는 차시를 늘이고, 주변 차시를 줄여 시수를 확보하고 수업을 재구성한다.</li> </ul>
3.2 융합 주제를 중심으로 융합 할 교과와 소프트웨어 수업의 내용이 하나의 맥락으로 통합될 수 있도록 수업을 재구성한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>예시 1) 초등학교 5~6학년 음악 : 프로그래밍 소프트웨어를 활용하여 악기를 이용한 악보를 만들고 연주해보기</li> <li>예시 2) 초등학교 6학년 실과 : ‘자원 관리와 자립’ 단원의 내용을 중심으로 쓰레기 분리수거를 홍보하는 애니메이션 만들기</li> </ul>
<b>4. 사용 용이성의 원리</b>	
교과융합 소프트웨어수업의 학습목표 달성을 효과적으로 지원하면서 사용 용이성을 가진 학습 도구 및 학습 자원을 지원한다.	
학습도구 및 정보자원	4.1 초보 학습자도 쉽게 사용할 수 있는 직관적 인터페이스의 프로그래밍 소프트웨어 도구를 선정한다.
예시 및	<ul style="list-style-type: none"> <li>프로그래밍 소프트웨어 : 엔트리</li> </ul>

해설	(https://playentry.org/), 스크래치 (https://scratch.mit.edu/) 등의 프로그래밍 언어를 쉽고 직관적으로 이해할 수 있는 블록형 프로그래밍 소프트웨어를 활용한다.
4.2 학습자들이 쉽게 이해하고 사용할 수 있는 사고 지원 도구를 선정한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 학습 환경과 자원을 고려하여 확보 및 적용이 적절하고 학습자가 쉽게 이해할 수 있는 발산적·수렴적 사고지원도구를 선정한다.</li> <li>• 예를 들어, 학습자가 오프라인 환경에서 활동지를 작성하는 활동에 익숙하다면, 마인드맵 활동지를 제시할 수 있는 브레인스토밍 기법과 아이디어 수렴을 위해 체크리스트 양식의 활동지를 제시할 수 있는 HIT 사고도구를 고려할 수 있다.</li> </ul>
<b>5. 정보자원 제공의 원리</b> 학습목표 달성을 위해 온라인 학습자료 및 소프트웨어를 선정하고 수업에 활용한다.	
5.1 창의적 문제해결 활동 및 프로젝트 구현을 위한 온라인 학습공간 및 웹기반 시스템을 제공한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 블록형 프로그래밍 소프트웨어 제공 사이트 : 엔트리에서 제공하는 학급 개설 및 공유 기능을 활용하거나, 스크래치에서 제공하는 스튜디오 기능을 활용한다.</li> <li>• Moodle, Canvas, Edmodo, 구글 클래스룸 등 학습관리시스템(LMS) : 학급용 계정을 개설한 후 학습자들의 결과물을 탑재한다.</li> </ul>
5.2 학습자료의 검색을 효율적으로 수행할 수 있도록 학습 내용과 관련한 검색 키워드 및 참고사이트, 자료 아카이브를 사전에 만들어 제공한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 지정된 시간 내에 효과적으로 주요한 자료를 검색하거나 찾을 수 있도록 검색 사이트, 검색할 수 있는 단어, 참고 사이트 및 주제에 대한 주요 내용을 정리하고 자료의 링크 주소 등을 정리한 자료 아카이브를 온라인 상에 게시하고 활동에 참고할 것을 안내한다.</li> </ul>

	<p>5.3 자료를 정리/조직하고 모둠원간 공유할 수 있도록 온라인 공동 작업 소프트웨어 프로그램을 활용한다.</p> <p>○ 온라인 공동작업 소프트웨어</p> <ul style="list-style-type: none"> <li>구글독스(<a href="https://docs.google.com/">https://docs.google.com/</a>) : 모둠별로 구글 문서를 생성하고 학생에게 파일 수정권한을 부여 후 주요 학습 내용과 자료를 링크하여 공유할 수 있도록 안내한다.</li> <li>패드렛(<a href="https://ko.padlet.com/">https://ko.padlet.com</a>) : 모둠별로 게시판을 생성하고 학습내용과 관련된 텍스트나 이미지 파일을 업로드하고 공유할 수 있도록 안내한다</li> </ul> <div data-bbox="475 786 576 869" style="text-align: center;"> <p>예시 및 해설</p> </div> <div data-bbox="735 831 1283 1211" style="text-align: center;"> </div> <p style="text-align: center;">&lt;패드렛 예시화면&gt;</p>
	<p>5.4 학습 과제 및 활동에 대해 분명한 준거(과제 수행 시간, 완성도, 평가 계획)에 대해 안내문을 작성하여 온라인 학습공간에 게시한다.</p>
<p style="text-align: center;">예시 및 해설</p>	<ul style="list-style-type: none"> <li>공지사항은 항상 열어볼 수 있도록 게시판을 별도로 생성하여 게시하거나 게시판 상위에 노출되도록 하고, 필요시 이미지나 참조링크를 함께 첨부한다.</li> <li>필요시 개별 학생에게 공지사항 알림 이메일이 발송되도록 설정한다.</li> </ul>
<p style="text-align: center;">창의적 교과-소프트웨어융합 설계활동 지원</p>	<p><b>6. 문제이해 지원의 원리</b></p> <p>학습자가 문제를 발견하고 이해할 수 있도록 학습 지원 자료와 질문을 제시한다.</p> <p>6.1 수업 내용과 관련된 실생활의 예시 자료(동영상, 애니메이션, 신문기사 등)을 제시하고 학습자의 경험을 공유하도록 안</p>

내한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 학습자가 실생활에서 경험하고 공감할 수 있는 친숙한 문제를 선정하고 참고할 만한 사이트링크 및 멀티미디어 콘텐츠를 온라인 플랫폼 상에 탑재한다.</li> <li>• 모둠 또는 개별적으로 자신의 경험과 생각을 온라인 플랫폼 상에 게시하여 서로 댓글을 작성하게 하거나, 발표를 통해 공유한다.</li> </ul>
6.2 학습자가 문제 상황을 이해하고 정의할 수 있도록 수업 주제, 해결해야 하는 문제, 해결 목적과 관련한 구체적인 질문을 제시한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 문제 상황에 대해 어떤 의문점이 드나요?</li> <li>• 문제 상황의 원인은 무엇일까요?</li> <li>• 문제와 관련된 이해당사자는 누구일까요?</li> <li>• 무엇이 문제인 걸까요?</li> <li>• 문제를 해결하기 위해 무엇부터 알아보아야 할까요?</li> <li>• 왜 이 문제를 해결해야 할까요?</li> </ul>
6.3 학습자 수준을 고려하여 최종적으로 도출할 수 있는 산출물의 예시를 제시한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 성취기준과 공통 주제, 수업 목표, 학습 내용을 고려하여 범위 내의 내용을 포함한 산출물을 작성할 수 있도록 안내한다.</li> <li>• 산출물 형태는 게임, 애니메이션, 시뮬레이션 등 다양한 형태로 제시될 수 있으나, 학습자 수준을 고려하여 사전에 구현이 가능한 과제를 수행하도록 피드백한다.</li> </ul>
<b>7. 아이디어 생성 지원의 원리</b>	
창의적 문제해결을 위한 아이디어를 발견할 수 있도록 발산적/수렴적 사고 지원 도구를 활용한다.	
7.1 문제에 대한 다양한 해결 아이디어를 도출할 수 있도록 발산적 사고도구(브레인스토밍, 속성열거법, SCAMPER)를 활용한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>○ 발산적 사고도구의 종류와 예시</li> <li>• 브레인스토밍 : 일정한 주제에 대하여 자유로운</li> </ul>

		<p>토론을 통해 아이디어를 생성해 내고자 할 때 사용하는 사고 기법</p> <ul style="list-style-type: none"> <li>• 속성열거법 : 문제상황, 물건, 과제의 속성을 열거한 후 이를 변화, 수정함으로써 각 속성별 아이디어를 도출하는 확장적 사고기법</li> <li>• SCAMPER : S(substitute, 대치하기), C(combine, 결합하기), A(adapt, 응용하기), M(modify - magnify - minify, 수정, 확대, 축소하기), P(put to other use, 새로운 용도), E(eliminate, 제거하기), R(rearrange-reverse, 재배열하기)의 약자로 아이디어 산출을 위해 사고의 출발점 혹은 문제해결의 착안점을 미리 항목별로 써 놓고 하나씩 체크하면서 아이디어를 내고자 하는 체크리스트 법(백연경, 2005)</li> </ul>
<p>7.2 문제에 대한 해결책을 찾을 수 있도록 수렴적 사고도구(HIT, PMI, 평가행렬법)를 활용한다.</p>		
예시 및 해설		<p>○ 수렴적 사고도구의 종류와 예시</p> <ul style="list-style-type: none"> <li>• HIT : 여러 가지 대안들 가운데서 가장 최적의 아이디어에 체크(V)표시를 하는 기법으로 좁혀야 할 대안의 수가 많을 때 사용함(임철일 외, 2014)</li> <li>• PMI : 하나의 아이디어의 분석 및 주의를 집중하는 도구로, P는 Plus로 제시된 아이디어의 좋은 점을, M은 Minus로 나쁜 점을, I는 Interesting으로 흥미로운 점을 나타냄(임철일 외, 2014)</li> <li>• 평가행렬법 : 여러 가지 아이디어를 평가 기준에 따라 각각의 강점과 약점을 파악하여 선택하는데 사용(임철일 외, 2014)</li> </ul>
<p>7.3 아이디어의 정리와 조직화를 위해 시각화 지원도구(종이, 포스트잇, 카드보드, 화이트보드)를 활용한다.</p>		
예시 및 해설		<ul style="list-style-type: none"> <li>• 포스트잇을 활용한 아이디어 정리</li> </ul>



<아이디어 정리 예시>

### 8. 논리화 지원의 원리

아이디어의 핵심요소를 파악하고 논리적인 해결안(알고리즘)을 작성할 수 있도록 지원한다.

8.1 중요한 아이디어를 선별하여 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호작용 등을 이해하고 해결안을 표현할 수 있도록 스토리보드 형태의 활동지를 제공한다.

예시 및  
해설

- 주요 장면에 대한 시각적 화면 구성과 논리 구성 (장면의 텍스트, 블록 코드의 논리적 배열과 순서)이 동시에 이루어지도록 활동지를 이등분하여 구성한다.

⇒ 만들고 싶은 프로그램의 장면 모습과 논리 구조를 설계하여 봅시다.

장면 생각하기	논리 생각하기
1)	1)
2)	2)
3)	3)

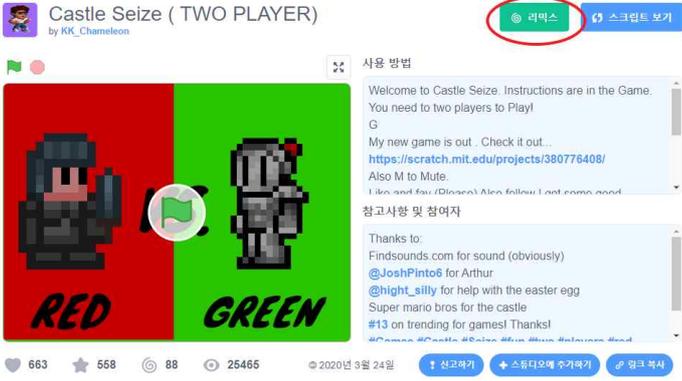
<활동지 예시>

8.2 알고리즘 작성 지원을 위한 학습자료(코딩 명령어 매뉴얼, 코딩 작성 예시자료)를 제시한다.

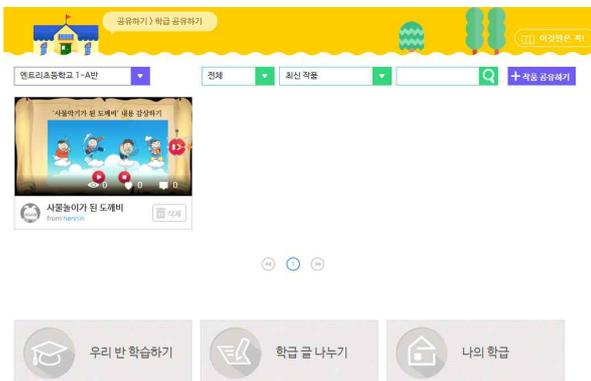
예시 및  
해설

- 프로그램 작성 방법과 사용하는 방법을 순서에 따라 구체적으로 설명한 해설 자료를 제공하기
- 블록별(ex. 무한반복, 소리내기, 조건문 등), 프로젝트 별(ex. 로봇청소기 움직이기, 쓰레기 분리수

	거 게임 만들기)로 예시 자료를 제공하기						
8.3 의사코드와 순서도의 작성 방법을 안내하고, 최종 산출물의 구현 가능성을 모듈별 논의를 통해 검토할 수 있도록 안내한다.							
예시 및 해설	<ul style="list-style-type: none"> <li>• 의사코드와 순서도 예시</li> </ul> <div data-bbox="667 707 1340 1160" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>◊ 라면을 만드는 알고리즘을 의사 코드와 순서도로 작성해 보자.</p>  <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">자연어</th> <th style="width: 33%;">의사 코드</th> <th style="width: 33%;">순서도</th> </tr> </thead> <tbody> <tr> <td>           ① 냄비에 물 500mL를 넣는다.            ② 물을 끓인다.            ③ 면이 끓으면            ④ 면을 넣는다.            ⑤ 스프를 넣는다.            ⑥ 3분 동안 물을 끓인다.         </td> <td>           시작            냄비 ← 물 500mL            물을 끓인다.            면이 끓으면            냄비 ← <input type="text"/>            냄비 ← <input type="text"/>            3분 동안 물을 끓인다.            끝         </td> <td> ) -&gt; Add soup (냄비 ← <input type="text"/>) -&gt; Boil 3 minutes (3분 동안 물을 끓인다) -&gt; End (끝)." data-bbox="728 408 838 518"/&gt; </td> </tr> </tbody> </table> </div> <ul style="list-style-type: none"> <li>- 산출물의 형태와 기능을 문장 형태로 표현하게 한다.</li> <li>- 작성한 문장을 절차적으로 나누어 자연어로 표현하게 한다.</li> <li>- 나누어진 문장을 의사코드나 순서도의 형태로 표현하게 한다.</li> <li>- 모듈별로 알고리즘이 프로그래밍을 통해 산출물로 구현 가능한지 검토할 수 있도록 안내한다.</li> </ul>	자연어	의사 코드	순서도	① 냄비에 물 500mL를 넣는다. ② 물을 끓인다. ③ 면이 끓으면 ④ 면을 넣는다. ⑤ 스프를 넣는다. ⑥ 3분 동안 물을 끓인다.	시작 냄비 ← 물 500mL 물을 끓인다. 면이 끓으면 냄비 ← <input type="text"/> 냄비 ← <input type="text"/> 3분 동안 물을 끓인다. 끝	) -> Add soup (냄비 ← <input type="text"/> ) -> Boil 3 minutes (3분 동안 물을 끓인다) -> End (끝)." data-bbox="728 408 838 518"/>
자연어	의사 코드	순서도					
① 냄비에 물 500mL를 넣는다. ② 물을 끓인다. ③ 면이 끓으면 ④ 면을 넣는다. ⑤ 스프를 넣는다. ⑥ 3분 동안 물을 끓인다.	시작 냄비 ← 물 500mL 물을 끓인다. 면이 끓으면 냄비 ← <input type="text"/> 냄비 ← <input type="text"/> 3분 동안 물을 끓인다. 끝	) -> Add soup (냄비 ← <input type="text"/> ) -> Boil 3 minutes (3분 동안 물을 끓인다) -> End (끝)." data-bbox="728 408 838 518"/>					
<b>9. 점진/반복적 문제해결 지원의 원리</b>							
<p>학습자가 스스로 계획하고 구현해 보는 활동을 통해 점진적, 반복적으로 아이디어와 문제 해결 방법을 개발할 수 있도록 지원한다.</p>							
<p>9.1 학습자가 점진적, 반복적으로 프로그램을 테스트하고 디버깅할 수 있도록 피드백을 제공한다.</p>							

예시 및 해설	<ul style="list-style-type: none"> <li>• 학습자가 프로그램을 실행하면서 코드의 오류를 발견하고 해결하거나 변수나 입력값, 출력값 등의 조절을 통해 최적의 산출물을 도출할 수 있도록 학습자의 진행 상태를 관찰하고 필요시 피드백을 제공한다.</li> <li>• 디버깅 : 오류가 발생한 알고리즘의 부분과 발생 원인을 찾아 해결하는 과정</li> </ul>
<p>9.2 학습자의 리믹스 활동을 지원하기 위해 프로그래밍의 구성요소를 일부 제공하거나, 동료나 교수자 및 온라인상의 다른 학습자들이 제작한 소프트웨어의 코드를 열람하고 이를 바탕으로 알고리즘을 발전시킬 수 있도록 안내한다.</p>	
예시 및 해설	 <p>&lt;스크래치에서 제공하는 리믹스 기능&gt;</p> <ul style="list-style-type: none"> <li>• 리믹스 : 기존 프로그래밍 코드를 수정하고 재사용하거나 프로그래밍 구성요소의 일부를 제공하여 이를 바탕으로 수정하는 것</li> </ul>
<p>9.3 컴퓨팅 사고력과 관련한 지식(반복, 연산자, 변수, 알고리즘 등)을 활용하여 점차적으로 복잡하고 어려운 코드를 작성할 수 있도록 난이도를 고려하여 수업을 설계한다.</p>	
예시 및 해설	<ul style="list-style-type: none"> <li>• 간단한 알고리즘으로 작성할 수 있는 문제 상황을 제시하고 점진적으로 복잡한 문제상황을 제시한다.</li> <li>• 수업 초반에서는 상대적으로 이해하기 쉬운 반복</li> </ul>

	명령어에 대해 안내하고, 후반에는 조건 명령어를 활용해 보도록 안내한다.
<b>10. 동료 프로그래밍 지원의 원리</b>	
각 학습자에게 역할을 부여하고 학습자 간 동료 프로그래밍 활동을 통해 상호협력적으로 산출물을 제작 할 수 있도록 모듈 별로 팀을 구성한다.	
10.1 컴퓨터를 활용한 프로그래밍 작업 경험이 풍부한 학습자들과 상대적으로 부족한 학습자들을 하나의 모듈로 구성한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>모든 학습자들이 비슷한 수준의 프로그래밍 경험을 가지고 있는 것이 아니므로, 다양한 수준의 학습자를 하나의 모듈로 구성하여 서로 도움을 주고받으며 경험이 풍부한 학습자가 기술적인 면에서 프로젝트 진행을 보완해 줄 수 있도록 지원한다.</li> </ul>
10.2 운전자 역할의 학습자와 네비게이터 역할의 학습자가 동시적·협력적으로 작업할 수 있도록 모듈별로 역할을 배정하고 각자의 역할에 대해 설명한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>○ 동료 프로그래밍(짝 프로그래밍) 적용하기 <ul style="list-style-type: none"> <li>두 학습자 또는 팀이 하나의 컴퓨터 앞에서 한 명은 운전자 역할을, 다른 한 명은 네비게이터 역할을 맡아 짝을 이루어 프로그래밍을 수행하도록 안내한다(4인 모듈 시 2인씩 역할 부여).</li> <li>운전자 역할 : 키보드와 마우스를 사용하여 타이핑을 하고 코드를 작성한다. 코딩 소프트웨어를 활용한 코드 작성 및 입력 등의 역할을 담당한다.</li> <li>네비게이터역할 : 프로젝트의 큰 그림에 집중하고 작성된 코드가 논리적인지 확인한다. 스토리보드 스케치, 화면 디자인, 아이디어 도출, 알고리즘 구상, 작성된 코드의 논리 및 오류 확인 등의 역할을 담당한다.</li> </ul> </li> </ul>
10.3 일정 기준을 중심으로 운전자와 네비게이터의 역할을 바꾸어 서로의 작업을 검토하고 보완할 수 있도록 안내한다.	
예시 및	<ul style="list-style-type: none"> <li>시간제한이나 해결해야 할 문제의 개수, 일정 수</li> </ul>

	해설	준의 진도, 기타 측정 가능한 기준을 만들어 이를 달성하면 바로 역할을 바꾸어 활동할 수 있도록 안내한다.
공유 및 평가 지원	<b>11. 공유 및 평가 지원의 원리</b>	
	학습자가 학습활동의 산출물을 제시·공유하여 평가 및 피드백을 받을 수 있는 기회를 제공한다.	
	11.1 학습자가 최종 산출물을 공유하고 배포할 수 있도록 공유 방법을 안내한다.	
	예시 및 해설	<ul style="list-style-type: none"> <li>엔트리(공유하기&gt;학급 공유하기&gt;작품공유하기 클릭)</li> </ul> 
	11.2 모둠 별 발표를 통해 산출물을 시연하고 의견을 나눌 수 있는 기회를 제공한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>학습자는 산출물을 만든 이유와 목적, 제작과정, 어려움과 느낀점을 발표하고 시연하며, 이에 대해 모둠 별로 자유롭게 의견을 제시한다. 교수자는 발표 및 논의 과정을 모니터링하고 필요 시 피드백을 제공한다.</li> </ul>	
11.3 학습자의 산출물을 다양한 관점에서 객관적으로 평가할 수 있도록 교수자 및 모둠 별 평가 기회를 제공한다.		
예시 및 해설	<ul style="list-style-type: none"> <li>교수자는 온라인 플랫폼 상에서 모둠 별 산출물에 대해 지원적이고 긍정적인 피드백을 중심으로 제공한다.</li> <li>학습자는 다른 모둠에서 발표한 내용에 대해 잘된 점과 개선할 점, 궁금한 점을 정리하여 온라인 플랫폼 상에 작성하도록 안내 한다.</li> <li>모둠별로 평가 결과를 확인하고 다른 모둠의 질문에 답변하고 개선할 수 있는 점을 작성하도록</li> </ul>	

	안내한다.
	11.4 학습자가 전체적인 학습활동을 돌이켜볼 수 있도록 구조화된 성찰일지 작성을 지원한다.
예시 및 해설	<ul style="list-style-type: none"> <li>○ 성찰일지 문항 예시</li> <li>• 문제를 해결하며 만들어낸 결과물이 마음에 드나요? 어떠한 점을 더 개선할 수 있을까요?</li> <li>• 여러분이 선택한 문제 해결 방법은 무엇이었나요? 그것이 가장 효과적인 방법이라고 생각하나요? 혹시 다른 문제해결 방법을 생각해 볼 수 있을까요?</li> <li>• 과학(수학)과 소프트웨어를 융합한 다양한 문제들과 해결방법에 대해 더 생각해 볼 수 있을까요?</li> <li>• 앞으로 문제를 해결해야 할 때 이번 활동으로 배운 것들을 적용할 수 있을까요?</li> </ul>
	11.5 학습자의 자유로운 사고를 촉진하고 학습자간 의견을 부담 없이 공유할 수 있는 열린 학습 분위기를 형성한다.
예시 및 해설	<ul style="list-style-type: none"> <li>• 산출물 평가 과정에서 학습자간 자유롭게 토의하고 의견을 제시할 수 있도록 촉진한다.</li> <li>• 오류발견을 통한 개선활동이 적절히 이루어지도록 실패에 긍정적이고 허용적인 분위기를 조성한다.</li> </ul>

## 나) 2차 전문가 타당화 결과

총 11개의 설계원리와 32개의 상세지침으로 구성된 2차 설계원리의 개별 설계원리에 대한 2차 전문가 타당화 결과는 <표 IV-13>와 같다. 개별 설계원리에 대한 타당화 점수의 평균은 3.60점~3.80점 사이에 분포하고 있는 것으로 나타난다. 또한 상세지침에 대한 평균은 3.00점~3.80점 사이에 분포하고 있는 것으로 나타난다. 따라서 2차 개별 설계원리와 상세지침은 비교적 타당한 것으로 해석할 수 있다. 개별 설계원리에 대한 CVI는 11개 설계원리 모두 1.00으로 분포하고 있어 개별 설계원리에 대한 평가자들의 의견이 일치하고 있는 것으로 나타났으며, 상세지침의 경

우 32개의 1개를 제외한 대부분의 상세지침이 0.8 이상으로 나타났다. IRA는 11개의 설계원리 중 8개가 1.00으로 나타났으며, 0.70~1.00 사이의 분포를 보이고 있어 대부분의 개별 설계원리가 신뢰할 만 하다고 해석할 수 있다.

<표 IV-13> 2차 개별 설계원리에 대한 2차 전문가 타당화 결과

구성요소	설계원리	전문가					평균	CVI	IRA
		B	C	F	G	H			
교과융합 교육과정	1. 적절성 확인의 원리	4	4	4	3	3	3.60	1.00	0.70
	2. 융합문제 설계의 원리	4	4	4	3	3	3.60	1.00	1.00
	3. 수업 재구성의 원리	4	4	4	3	3	3.60	1.00	1.00
학습도구 및 정보자원	4. 사용 용이성의 원리	4	4	4	3	3	3.60	1.00	0.70
	5. 정보자원 제공의 원리	4	4	4	3	4	3.80	1.00	0.80
창의적 교과-소 프트웨어 융합 설계활동 지원	6. 문제이해 지원의 원리	4	4	4	3	4	3.80	1.00	1.00
	7. 아이디어 생성 지원의 원리	4	4	4	3	3	3.60	1.00	1.00
	8. 논리화 지원의 원리	4	4	4	3	4	3.80	1.00	1.00
	9. 점진/반복적 문제 해결 지원의 원리	4	3	4	3	4	3.60	1.00	1.00
	10. 동료 프로그래밍 지원의 원리	4	4	4	3	3	3.60	1.00	1.00
공유 및 평가 지원	11. 공유 및 평가 지원의 원리	4	4	4	3	4	3.80	1.00	0.80

2차 전문가 타당화를 통해 수집한 개별 설계원리 및 상세지침에 대한 의견과 개선점을 정리하면 <표 IV-14>와 같다. 첫째, 설계원리와 상세지침의 내용과 서술방식의 일부 모호한 표현을 구체적으로 수정하는 것이 필요하다. 둘째, 설계원리와 상세지침에 제시된 해설 중 일부는 구체적인 활용 예시가 필요하여 이에 대한 수정이 필요하다. 셋째, 구성요소, 개별원리와 상세지침의 위계가 맞지 않는 부분이 있어 위계에 맞도록 일부를 추가, 삭제하고 위치를 조정하여 재구성할 필요가 있다. 마지막으로, 상세지침을 아우를 수 있는 개별 원리를 제시해야 하며, 이를 위해 용어와 표현을 명확히 하여 개별 원리 명칭을 수정할 필요가 있다.

<표 IV-14> 2차 전문가 타당화 검토 의견과 수정 및 개선사항

전문가 의견 및 수정점		세부 개선사항
재구조화	구성요소 재구조화	<ul style="list-style-type: none"> <li>▪ 교과-소프트웨어 융합 교육과정, 소프트웨어 학습도구, 교과-소프트웨어 융합 학습지원, 교과-소프트웨어 융합 수업활동, 공유 및 평가로 구성요소 재구조화</li> </ul>
	원리 및 상세지침 추가 및 삭제	<ul style="list-style-type: none"> <li>▪ 원리 4, 원리 6 추가</li> <li>- 상세지침 1.1, 1.4, 4.1 추가</li> <li>- 상세지침 2.1, 11.1 삭제</li> </ul>
	원리 및 상세지침 위치 조정	<ul style="list-style-type: none"> <li>▪ 1차 원리 10을 2차 원리 7로 위치 조정</li> <li>▪ 1차 원리 6을 2차 원리 8로 위치 조정</li> <li>▪ 1차 원리 7을 2차 원리 9로 위치 조정</li> <li>▪ 1차 원리 8을 2차 원리 10으로 위치 조정</li> <li>▪ 1차 원리 9를 2차 원리 11로 위치 조정</li> <li>- 상세지침 5.1 → 4.2로 위치 조정</li> <li>- 11.5 → 6.3</li> <li>- 10.1, 10.2, 10.3 → 7.1, 7.2, 7.3</li> <li>- 6.2, 6.3 → 8.2, 8.3</li> </ul>

		<ul style="list-style-type: none"> <li>- 7.1, 7.2, 7.3 → 9.1, 9.2, 9.3</li> <li>- 8.1, 8.3 → 10.1, 10.2</li> <li>- 8.2 → 6.1</li> <li>- 9.1, 9.2, 9.3 → 11.1, 11.2, 11.3</li> </ul>
	원리 및 상세지침 통합 및 세분화	<ul style="list-style-type: none"> <li>▪ 1차 원리 5를 2차 원리 5와 6으로 세분화</li> <li>- 1.1 → 1.1, 1.2 세분화</li> <li>- 1.2 → 1.3, 1.4 세분화</li> </ul>
표현의 구체화 및 명확화	내용 진술 구체화	<ul style="list-style-type: none"> <li>▪ 내용 진술 구체화</li> <li>- 1.2 → 1.3의 내용 구체화</li> <li>- 4.1 → 5.1의 내용 진술 구체화</li> <li>▪ 동사 표현 구체화</li> <li>- 2.1, 2.2, 5.1, 11.3, 12.2 동사 표현 구체화</li> </ul>
	진술형태 수정	<ul style="list-style-type: none"> <li>▪ 서술형태(~한다.)를 명령형태(~하라.)로 수정</li> </ul>
	용어 변경	<ul style="list-style-type: none"> <li>▪ ‘적절성 확인의 원리’ → ‘수업 실행성의 원리’ 로 변경</li> <li>▪ ‘수업 재구성의 원리’ → ‘맥락 재구성의 원리’ 로 변경</li> <li>▪ ‘융합 문제 설계의 원리’ → ‘실제적 문제의 원리’ 로 변경</li> <li>▪ ‘동료 프로그래밍 지원의 원리’ → ‘프로그래밍 모듈 지원의 원리’ 로 변경</li> <li>▪ ‘점진/반복적 문제해결 지원의 원리’ → ‘점진적 문제해결 지원성의 원리’ 로 변경</li> </ul>

위와 같이 제안된 수정사항에 따라 총 12개의 설계원리와 32개의 상세지침을 도출하였다. 3차 설계원리 및 상세지침은 <표 IV-15>와 같다.

<표 IV-15> 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계원리와 상세지침 - 3차

구성요소	원리 및 상세지침	
교과-소프트웨어융합 교육과정	<b>1. 수업 실행성의 원리</b> 소프트웨어와 연계할 학습내용과 수업 환경을 분석하고 교과-소프트웨어융합 수업의 실행 가능성과 효과성을 판단한다.	
	1.1 융합하려는 교과 및 단원의 성취기준, 학습내용, 교수학습 방법, 평가방법을 확인하고 교과-소프트웨어융합수업이 구현 가능한지 판단하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 융합할 교과목과 소프트웨어융합수업을 적용시킬 단원은 무엇인가?</li> <li>• 단원의 성취기준과 학습내용은 무엇인가?</li> <li>• 수업활동과 산출물에 대한 평가도구 및 평가방법은 무엇인가?</li> </ul>
	1.2 학습자 수준을 파악하여 학습자가 교과-소프트웨어융합 내용을 이해하고 산출물을 만들 수 있는지 확인하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 학습자는 컴퓨팅 사고력이 무엇인지 알고 있는가?</li> <li>• 학습자는 수업 내용과 과제를 이해하고 산출물을 만들어낼 수 있는가?</li> <li>• 학습자의 컴퓨터 프로그래밍 소프트웨어 조작 수준은 어느 정도인가?</li> <li>• 학습자는 교과-소프트웨어융합 수업에 대한 선행 경험이 있는가?</li> </ul>
1.3 수업의 실행 및 지원 가능성을 확인하기 위해 학습 환경, 물리적 자원(HW), 학교 방침 및 분위기, 전문가 그룹 등 제반 자원과 환경을 파악하라.		

예시 및 해설	<ul style="list-style-type: none"> <li>• 학교 방침과 문화가 교과-소프트웨어융합수업을 실행하기에 적절한가?</li> <li>• 교과-소프트웨어융합수업 실행을 위해 동료 교사의 이해와 협조가 적절히 이루어지는가?</li> <li>• 교과-소프트웨어융합수업 실행을 위해 시간과 자원, 비용이 얼마나 소요되고, 이를 충당할 수 있는가?</li> <li>• 교과-소프트웨어융합수업을 설계하고 실행하기 위해 조언을 받을 수 있는 전문가 그룹(동료 교사, 프로그래밍 전문가)이 있는가?</li> </ul>
1.4 프로그래밍 소프트웨어를 활용한 교과융합 수업이 성취 기준 달성에 도움이 되는지를 확인하라.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 교과-소프트웨어융합 수업이 기존 교과 수업보다 상대적으로 어떤 이점이 있는가?</li> <li>• 프로그래밍 소프트웨어를 활용하는 것이 활용하지 않는 수업 보다 어떤 점에서 효과적인가?</li> <li>• 교과-소프트웨어융합수업이 학습자들의 학업 성취에 도움이 되는가?</li> </ul>
<b>2. 맥락 재구성의 원리</b> 융합할 교과/단원의 학습내용을 바탕으로 교과-소프트웨어융합 교육과정을 하나의 맥락으로 재구성한다.	
2.1 교과 특성, 주제의 연계성, 학습자 수준을 고려하여 주제와 관련된 주요 내용을 중심으로 필요시 내용을 생략하거나 단원의 순서를 바꾸도록 하라.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 각 교과의 성취기준을 확인하고 공통 주제를 선정 한 후, 통합된 학습 내용을 중심으로 교육과정 상의 수업 순서를 바꾸거나 주요 내용을 다루는 차시를 늘이고, 주변 차시를 줄여 시수를 확보하고 수업을 재구성한다.</li> </ul>

2.2 융합 주제를 중심으로 융합 할 교과와 소프트웨어 수업의 내용을 하나의 맥락으로 통합하라.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 예시 1) 초등학교 5~6학년 음악 : 프로그래밍 소프트웨어를 활용하여 악보를 만들고 연주 해보기</li> <li>• 예시 2) 초등학교 6학년 실과 : ‘자원 관리와 자립’ 단원의 내용을 중심으로 쓰레기 분리 수거를 홍보하는 애니메이션 만들기</li> </ul>
<b>3. 실제적 문제의 원리</b>	
융합 과목의 소프트웨어 연계요소를 고려하여 학습자에게 친근감 있고 도전감 있는 실제적인 교과융합 문제를 설계한다.	
3.1 융합주제, 수업목표, 과제 형식을 고려하여 컴퓨팅 사고를 활용하여 해결할 수 있는 실제 생활의 문제를 선정하라.	
예시 및 해설	<p>○ [6사05-03] 초등학교 6학년 사회과목의 ‘우리나라의 정치 발전’ 단원 중 ‘일상생활과 민주주의’</p> <ul style="list-style-type: none"> <li>• 성취기준 : 일상생활에서 경험하는 민주주의 실천 사례를 탐구하여 민주주의의 의미와 중요성을 파악하고, 생활 속에서 민주주의를 실천하는 태도를 기른다.</li> <li>• 교과-소프트웨어융합주제 : 민주주의 실천을 위한 전자 투표 프로그램 만들어 보기</li> <li>• 교과-소프트웨어융합학습 목표 : 1) 전자투표 프로그램을 설계할 수 있다. 2) 리스트와 조건문을 활용하여 프로그램을 만들 수 있다. 3) 만든 전자투표 프로그램을 활용하여 투표를 실시할 수 있다.</li> </ul> <p>(엔트리 학습자료 ‘전자투표 프로그램으로 모의선거 실시하기’ 자료 참고)</p>
3.2 학습자가 친숙함을 느낄 수 있는 시나리오 형태로 문제를 제시하라.	

	예시 및 해설	<ul style="list-style-type: none"> <li>[6실03-04] 초등학교 6학년 실과 과목의 ‘자원 관리와 자립’ 단원 내용을 중심으로 환경오염으로 발생하는 생활 주변의 문제점들을 알아보고, 쓰레기 재활용을 통한 환경 보호의 필요성을 안내 후, 분리수거와 관련된 문제를 해결할 수 있는 프로그램 설계하기 (엔트리 학습자료 ‘자동으로 분리수거를 해요’ 자료 참고)</li> </ul>
	3.3 해결안이 하나 이상으로 제시될 수 있는 복합적이고 비구조화된 형태의 문제를 설계하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>비구조화 형태의 문제 설계 시 고려할 점 : 1) 문제와 관련된 상황이나 요소가 분명히 정의되어 있지 않은 문제로 설계한다. 2) 문제 해결에 필요한 정보가 충분히 포함되어 있지 않은 문제를 설계한다. 3) 다양한 해답이나 해결 경로를 가지고 있다. 4) 의사결정의 명료화와 논쟁에 초점을 둔다.</li> </ul>
소프트웨어 학습도구	<b>4. 소프트웨어도구 적합성의 원리</b>	
	효과적인 수업 구성 및 수업 목표 달성을 위해 적합한 소프트웨어도구를 수업 활동의 목적에 맞게 선정하고 활용한다.	
	4.1 계획된 최종 산출물 형태를 효과적으로 구현하기에 적합한 소프트웨어도구를 선정하고 사전에 설치를 진행하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>프로그래밍 소프트웨어 : 엔트리(<a href="https://playentry.org/">https://playentry.org/</a>), 스크래치(<a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a>)</li> <li>오픈소스 HW : 아두이노, 마이크로비트, 라즈베리파이</li> </ul>
4.2 창의적 문제해결 활동 및 프로젝트 구현을 위한 온라인 학습공간 및 LMS 시스템을 제공하라.		

	예시 및 해설	<ul style="list-style-type: none"> <li>• 블록형 프로그래밍 소프트웨어 제공 사이트 : 엔트리에서 제공하는 학급 개설 및 공유 기능을 활용하거나, 스크래치에서 제공하는 스튜디오 기능을 활용한다.</li> <li>• 학습관리시스템(LMS) : 학급용 계정을 개설한 후 학습자들의 결과물을 탑재한다. ex) 구글 클래스룸, e-학습터, Moodle, Canvas, Edmodo</li> </ul>
<b>5. 사용 용이성의 원리</b> 교과융합 소프트웨어수업의 학습목표를 효과적으로 달성하기 위해사용 용이성을 가진 학습 도구를 지원한다		
5.1 초보 학습자도 쉽게 사용할 수 있는 직관적 인터페이스의 프로그래밍 소프트웨어 도구를 선정하고 학습자들이 가입할 수 있도록 사전에 안내하라.		
	예시 및 해설	쉽고 직관적으로 프로그래밍을 이해할 수 있도록 엔트리( <a href="https://playentry.org/">https://playentry.org/</a> ), 스크래치( <a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a> ) 등의 EPL 언어를 활용한다.
5.2 학습자들이 쉽게 이해하고 사용할 수 있는 소프트웨어기반 사고 지원 도구를 선정하라.		
	예시 및 해설	<ul style="list-style-type: none"> <li>• 학습 환경과 자원을 고려하여 확보 및 적용이 적절하고 학습자가 쉽게 이해할 수 있는 발산적·수렴적 사고 지원도구를 선정한다.</li> <li>• 마인드마스터, S3CPS 등의 창의적 문제해결을 위한 웹기반 사고 지원도구를 활용한다.</li> </ul>
교과-소	<b>6. 자료의 명료성의 원리</b>	

소프트웨어 융합 학습지원	학습내용과 학습 활동에 대한 안내 자료와 규칙을 알기 쉽고 명료한 형태로 제작하여 제공한다.	
	6.1 학습 내용과 관련한 검색 키워드 및 참고사이트, 자료 아카이브, 학습자료(코딩 명령어 매뉴얼, 코딩 작성 예시자료)를 사전에 만들어 제공하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 지정된 시간 내에 효과적으로 주요한 자료를 검색하거나 찾을 수 있도록 검색 사이트, 검색할 수 있는 단어, 참고 사이트 및 주제에 대한 주요 내용을 정리하고 자료의 링크 주소 등을 정리한 자료 아카이브를 온라인 상에 게시하고 활동에 참고할 것을 안내한다.</li> <li>• 프로그램 작성 방법과 사용하는 방법을 순서에 따라 구체적으로 설명한 해설 자료를 제공한다.</li> </ul>
	6.2 수업 진행방식, 학습 과제 및 활동에 대한 분명한 준거(과제 수행 시간, 완성도, 평가 계획)에 대해 안내문을 작성하여 온라인 학습공간에 게시하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 공지사항은 항상 열어볼 수 있도록 게시판을 별도로 생성하여 게시하거나 게시판 상위에 노출되도록 하고, 필요시 이미지나 참조링크를 함께 첨부한다.</li> <li>• 필요시 개별 학생에게 공지사항 알림 이메일이 발송되도록 설정한다.</li> </ul>
	6.3 학습자의 자유로운 사고를 촉진하고 학습자간 의견을 부담없이 공유할 수 있는 열린 학습 분위기 형성을 위해 규칙과 방법을 계획하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 산출물 평가 과정에서 학습자간 자유롭게 토의하고 의견을 제시할 수 있도록 촉진한다.</li> <li>• 실패에 긍정적이고 허용적인 분위기 조성을 위해 규칙과 방법을 제시한다(ex. 마인드맵</li> </ul>

	소프트웨어를 활용하여 친구와 자유롭게 아이디어를 나누기, 나와 다른 아이디어에 대한 장점을 찾아보고 리스트를 만들기, 다른 모둠이 제작한 프로그램의 잘된 점에 대해 한 줄 칭찬 남기기).
<b>7. 프로그래밍 모둠 구성의 원리</b>	
각 학습자에게 역할을 부여하고 학습자 간 동료 프로그래밍 활동을 통해 협력적으로 작업할 수 있는 기회를 제공한다.	
7.1 컴퓨터를 활용한 프로그래밍 작업 경험이 풍부한 학습자들과 상대적으로 부족한 학습자들을 하나의 모둠으로 구성하라.	
예시 및 해설	<ul style="list-style-type: none"> <li>모든 학습자들이 비슷한 수준의 프로그래밍 경험을 가지고 있는 것이 아니므로, 다양한 수준의 학습자를 하나의 모둠으로 구성하여 서로 도움을 주고받으며 경험이 풍부한 학습자가 기술적인 면에서 프로젝트 진행을 보완해 줄 수 있도록 지원한다.</li> </ul>
7.2 운전자 역할의 학습자와 네비게이터 역할의 학습자가 동시적·협력적으로 작업할 수 있도록 모둠별로 역할을 배정하고 각자의 역할에 대해 설명하라.	
예시 및 해설	<ul style="list-style-type: none"> <li>○ 동료 프로그래밍(짝 프로그래밍) 적용하기</li> <li>• 두 학습자 또는 팀이 하나의 컴퓨터 앞에서 한 명은 운전자 역할을, 다른 한 명은 네비게이터 역할을 맡아 짝을 이루어 프로그래밍을 수행하도록 안내한다(4인 모둠 시 2인씩 역할 부여).</li> <li>• 운전자 역할 : 키보드와 마우스를 사용하여 타이핑을 하고 코드를 작성한다. 코딩 소프트웨어를 활용한 코드 작성 및 입력 등의 역할을 담당한다.</li> <li>• 네비게이터 역할 : 프로젝트의 큰 그림에 집중하고 작성된 코드가 논리적인지 확인한다. 스</li> </ul>

		토리보드 스케치, 화면 디자인, 아이디어 도출, 알고리즘 구상, 작성된 코드의 논리 및 오류 확인 등의 역할을 담당한다.
	7.3 일정 기준을 중심으로 운전자와 네비게이터의 역할을 바꾸어 서로의 작업을 검토하고 보완할 수 있도록 안내하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 시간 제한이나 해결해야 할 문제의 개수, 일정 수준의 진도, 기타 측정 가능한 기준을 만들어 이를 달성하면 바로 역할을 바꾸어 활동할 수 있도록 안내한다.</li> </ul>
교과-소 소프트웨어 융합 수업활동	<b>8. 문제이해 지원성의 원리</b> 학습자가 문제를 발견하고 이해할 수 있도록 학습 지원 자료와 질문프롬프트를 제시한다.	
	8.1 수업 내용과 관련된 실생활의 예시 자료(동영상, 애니메이션, 신문기사 등)를 제시하고 학습자의 경험을 공유하도록 안내하라.	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 학습자가 실생활에서 경험하고 공감할 수 있는 친숙한 문제를 선정하고 참고할 만한 사이트 링크 및 멀티미디어 콘텐츠를 온라인 플랫폼 상에 탑재한다.</li> <li>• 모듈 또는 개별적으로 자신의 경험과 생각을 온라인 플랫폼 상에 게시하여 서로 댓글을 작성하게 하거나, 발표를 통해 공유한다.</li> </ul>
	8.2 학습자가 문제를 이해하고 정의할 수 있도록 수업 주제, 해결해야 하는 문제, 해결 목적과 관련한 구체적인 질문을 제시하라.	

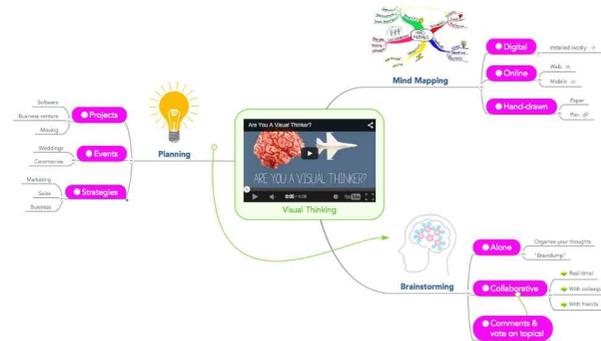
	예시 및 해설	<ul style="list-style-type: none"> <li>• 문제에 대해 어떤 의문점이 드나요?</li> <li>• 문제의 원인은 무엇일까요?</li> <li>• 문제와 관련된 이해당사자는 누구일까요?</li> <li>• 무엇이 문제인 걸까요?</li> <li>• 문제를 해결하기 위해 무엇부터 알아보아야 할까요?</li> <li>• 왜 이 문제를 해결해야 할까요?</li> </ul>
8.3 학습자 수준을 고려하여 최종적으로 도출할 수 있는 산출물의 예시를 제시하라.		
	예시 및 해설	<ul style="list-style-type: none"> <li>• 성취기준과 공통 주제, 수업 목표, 학습 내용을 고려하여 범위 내의 내용을 포함한 산출물을 작성할 수 있도록 안내한다.</li> <li>• 산출물 형태는 게임, 애니메이션, 시뮬레이션 등 다양한 형태로 제시될 수 있으나, 학습자 수준을 고려하여 사전에 구현이 가능한 과제를 수행하도록 피드백한다.</li> </ul>
<b>9. 아이디어 생성 지원성의 원리</b>		
창의적 문제해결을 위한 아이디어를 발견할 수 있도록 발산적/수렴적 사고 지원 도구를 활용한다.		
9.1 문제에 대한 다양한 해결 아이디어를 도출할 수 있도록 발산적 사고도구(브레인스토밍, 속성열거법, SCAMPER)를 활용하라.		
	예시 및 해설	<ul style="list-style-type: none"> <li>○ 발산적 사고도구의 종류와 예시</li> <li>• 브레인스토밍 : 일정한 주제에 대하여 자유로운 토론을 통해 아이디어를 생성해 내고자 할 때 사용하는 사고 기법</li> <li>• 속성열거법 : 문제상황, 물건, 과제의 속성을 열거한 후 이를 변화, 수정함으로써 각 속성 별 아이디어를 도출하는 확장적 사고기법</li> </ul>

		<ul style="list-style-type: none"> <li>• SCAMPER : S(substitute, 대치하기), C(combine, 결합하기), A(adapt, 응용하기), M(modify - magnify - minify, 수정, 확대, 축소하기), P(put to other use, 새로운 용도), E(eliminate, 제거하기), R(rearrange-reverse, 재배열하기)의 약자로 아이디어 산출을 위해 사고의 출발점 혹은 문제해결의 착안점을 미리 항목별로 써 놓고 하나씩 체크하면서 아이디어를 내고자 하는 체크리스트 법(백연경, 2005)</li> </ul>
	<p>9.2 문제에 대한 해결책을 찾을 수 있도록 수렴적 사고도구(HIT, PMI, 평가행렬법)를 활용하라.</p>	
	<p>예시 및 해설</p>	<ul style="list-style-type: none"> <li>○ 수렴적 사고도구의 종류와 예시</li> <li>○ HIT : 여러 가지 대안들 가운데서 가장 최적의 아이디어에 체크(V)표시를 하는 기법으로 좁혀가야 할 대안의 수가 많을 때 사용함(임철일 외, 2014)</li> <li>○ PMI : 하나의 아이디어의 분석 및 주의를 집중하는 도구로, P는 Plus로 제시된 아이디어의 좋은 점을, M은 Minus로 나쁜 점을, I는 Interesting으로 흥미로운 점을 나타냄(임철일 외, 2014)</li> <li>○ 평가행렬법 : 여러 가지 아이디어를 평가 기준에 따라 각각의 강점과 약점을 파악하여 선택하는데 사용(임철일 외, 2014)</li> </ul>
	<p>9.3 아이디어의 정리와 조직화를 위해 시각화 지원도구(패들릿, 웹기반 브레인스토밍 툴 등)를 활용하라.</p>	
	<p>예시 및 해설</p>	<ul style="list-style-type: none"> <li>• 패들릿(<a href="https://ko.padlet.com">https://ko.padlet.com</a>) : 모듈별로 게시판을 생성하고 학습내용과 관련된 텍스트나 이미지파일을 업로드하고 공유할 수 있도록 안내한다.</li> </ul>



<패들릿 예시화면>

- 마인드마스터(<https://www.mindmeister.com/ko>) : 웹기반 브레인스토밍 도구를 사용하여 다양한 아이디어와 단어를 효율적으로 정리할 수 있도록 안내한다.



## 10. 논리화 지원성의 원리

아이디어의 핵심요소를 파악하고 논리적인 해결안(알고리즘)을 작성할 수 있도록 지원한다.

10.1 중요한 아이디어를 선별하여 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호작용 등을 이해하고 해결안을 표현할 수 있도록 스토리보드 형태의 활동지를 제공하라.

예시 및  
해설

- 주요 장면에 대한 시각적 화면 구성과 논리 구성(장면의 텍스트, 블록 코드의 논리적 배열과 순서)이 동시에 이루어지도록 활동지를 이등분하여 구성한다.

☞ 만들고 싶은 프로그램의 장면 모습과 논리 구조를 설계하여 봅시다.

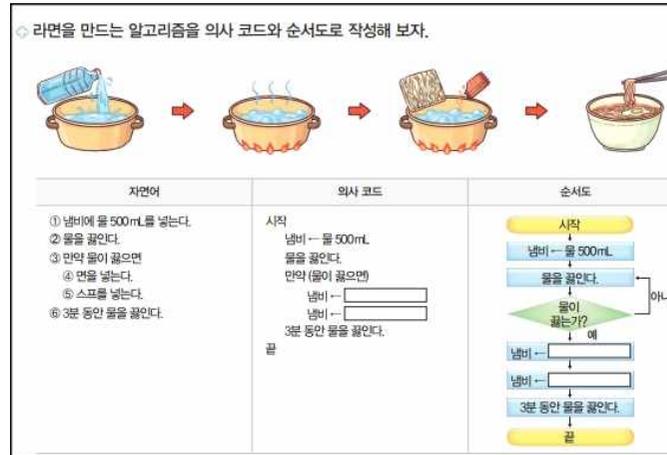
장면 생각하기	논리 생각하기
1)	1)
2)	2)
3)	3)

[활동지 예시]

10.2 의사코드와 순서도의 작성 방법을 안내하고, 작성한 알고리즘이 최종 산출물로 구현될 수 있는지 모듈별 논의를 통해 검토할 수 있도록 안내하라.

예시 및  
해설

○ 의사코드와 순서도 예시



- 산출물의 형태와 기능을 문장 형태로 표현하게 한다.
- 작성한 문장을 절차적으로 나누어 자연어로 표현하게 한다.
- 나누어진 문장을 의사코드나 순서도의 형태로 표현하게 한다.
- 모듈별로 알고리즘이 프로그래밍을 통해 산출물로 구현 가능한지 검토할 수 있도록 안내한다.

## 11. 점진적 문제해결 지원성의 원리

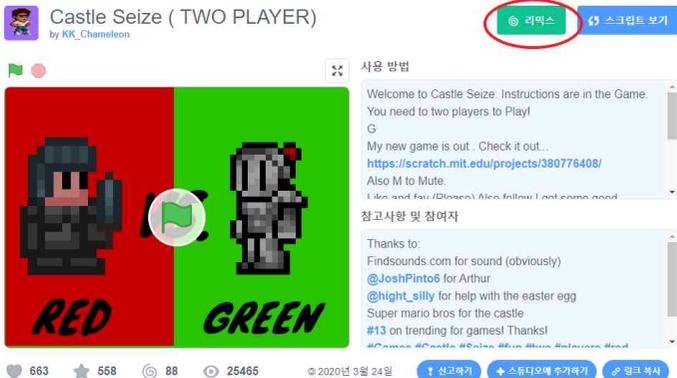
학습자가 스스로 계획하고 구현해 보는 활동을 통해 점진적으로 아이디어와 문제 해결 방법을 개발할 수 있도록 지원한다.

11.1 학습자가 점진적으로 프로그램을 테스트하고 디버깅할 수 있도록 피드백을 제공하라.

- 예시 및 해설
- 학습자가 프로그램을 실행하면서 코드의 오류를 발견하고 해결하거나 변수나 입력값, 출력값 등의 조절을 통해 최적의 산출물을 도출할 수 있도록 학습자의 진행상태를 관찰하고 필요시 피드백을 제공한다.
  - 디버깅 : 오류가 발생한 알고리즘의 부분과 발생 원인을 찾아 해결하는 과정

11.2 학습자의 리믹스 활동을 지원하기 위해 프로그래밍의 구성요소를 일부 제공하거나, 동료나 교수자 및 온라인상의 다른 학습자들이 제작한 소프트웨어의 코드를 열람하고 이를 바탕으로 알고리즘을 발전시킬 수 있도록 안내하라.

예시 및 해설



<스크래치에서 제공하는 리믹스 기능>

- 리믹스 : 기존 프로그래밍 코드를 수정하고 재사용하거나 프로그래밍 구성요소의 일부를

	제공하여 이를 바탕으로 수정하는 것
	11.3 컴퓨팅 사고력과 관련한 지식(반복, 연산자, 변수, 알고리즘 등)을 활용하여 점차적으로 복잡하고 어려운 코드를 작성할 수 있도록 난이도를 고려하여 활동순서를 조정하라.
예시 및 해설	<ul style="list-style-type: none"> <li>• 간단한 알고리즘으로 작성할 수 있는 문제 상황을 제시하고 점진적으로 복잡한 문제상황을 제시한다.</li> <li>• 수업 초반에서는 상대적으로 이해하기 쉬운 반복 명령어에 대해 안내하고, 후반에는 조건 명령어를 활용해 보도록 안내한다.</li> </ul>
공유 및 평가	<b>12. 공유 및 평가 지원성의 원리</b> 학습자가 학습활동의 산출물을 제시·공유하여 평가 및 피드백을 받을 수 있는 기회를 제공한다.
	12.1 모듈 별 발표를 통해 최종산출물을 시연하고 의견을 나눌 수 있는 기회를 제공하라.
	예시 및 해설 <ul style="list-style-type: none"> <li>• 학습자는 산출물을 만든 이유와 목적, 제작과정, 어려움과 느낀점을 발표하고 시연하며, 이에 대해 모듈 별로 자유롭게 의견을 제시한다. 교수자는 발표 및 논의 과정을 모니터링하고 필요 시 피드백을 제공한다.</li> </ul>

12.2 학습자의 산출물을 다양한 관점에서 객관적으로 평가할 수 있도록 모듈 별 평가 도구를 제공한다.	
예시 및 해설	<ul style="list-style-type: none"> <li>• 교수자는 온라인 플랫폼 상에서 모듈 별 산출물에 대해 지원적이고 긍정적인 피드백을 중심으로 제공한다.</li> <li>• 다른 모듈에서 발표한 내용에 대해 동료평가의 예시나 잘된 점과 개선할 점, 궁금한 점 등 평가요소가 포함된 평가도구를 제공한다.</li> <li>• 모듈별로 평가 결과를 확인하고 다른 모듈의 질문에 답변하고 개선할 수 있는 점을 작성하도록 안내 한다.</li> </ul>
12.3 학습자가 전체적인 학습활동을 돌이켜볼 수 있도록 구조화된 성찰일지 작성을 지원한다.	
예시 및 해설	<p>○ 성찰일지 문항 예시</p> <ul style="list-style-type: none"> <li>• 문제를 해결하며 만들어낸 결과물이 마음에 드나요? 어떠한 점을 더 개선할 수 있을까요?</li> <li>• 여러분이 선택한 문제 해결 방법은 무엇이었나요? 그것이 가장 효과적인 방법이라고 생각 하나요? 혹시 다른 문제해결 방법을 생각해 볼 수 있을까요?</li> <li>• 과학(수학)과 소프트웨어를 융합한 다양한 문제들과 해결방법에 대해 더 생각해 볼 수 있을까요?</li> <li>• 앞으로 문제를 해결해야 할 때 이번 활동으로 배운 것들을 적용할 수 있을까요?</li> </ul>

## 다. 개념모형에 관한 전문가 타당화

### 1) 1차 전문가 타당화 결과

개념모형에 대한 전문가 타당화는 모형의 타당성, 설명력, 유용성, 보편성, 이해도 측면에서 이루어졌다. 타당성은 도출한 개념모형이 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 참고할 수 있는 모형으로 타당한지에 대한 검토항목이다. 설명력은 도출한 개념모형으로 수업 설계 시 고려해야 할 요소를 잘 설명하고 있는지에 대한 것이며, 유용성은 개념모형이 수업 설계 시 유용하게 활용될 수 있는지를 묻는 항목이다. 보편성은 개념모형이 실제 학교 현장에서 보편적으로 이용할 수 있는지를, 이해도는 모형에 대한 표현들이 쉽게 이해할 수 있는지를 묻는 것이다.

초기 개념모형에 대한 1차 전문가 타당화 결과는 다음 <표 IV-16>와 같다.

<표 IV-16> 초기 개념모형에 대한 1차 전문가 타당화 결과

영역	전문가					평균	표준 편차	CVI	IRA
	A	B	C	D	E				
타당성	2	2	3	1	2	2.60	.548	0.20	
설명력	3	2	3	1	2	2.00	.000	0.40	
유용성	3	2	3	1	2	2.80	.447	0.40	0.00
보편성	2	2	2	1	3	1.20	.447	0.20	
이해도	3	2	3	2	3	2.40	.548	0.40	

초기 모형에 대한 1차 전문가 검토 결과는 타당성 평균 2.6점, 설명력 평균 2.0점, 유용성 평균 2.8점, 보편성 평균 1.2점, 이해도 평균 2.4점으로 나타나 비교적 낮은 타당도를 보였다. 각 문항에 대한 CVI는 0.2~0.4 사이에 분포하고 있고, IRA는 모든 문항에서 0.00으로 나타나 전체적인

모형의 수정 및 보완이 필요한 것으로 해석할 수 있다.

1차 전문가 타당화 결과를 통해 수집된 전문가 의견을 정리하면 다음과 같다. 첫째, 모형에 사용한 그림(메타포)이 의도를 명확히 설명하지 못하며, 각 원리 사이의 관계와 구조를 발견하기 어렵다. 둘째, 초기 개념모형의 구성요소 중 ‘안내’는 다른 구성요소인 ‘CT기반 창의적 융합설계활동 설계’, ‘교과융합 문제’, ‘협력적 학습 환경’, ‘학습도구(창의적 사고 도구, 컴퓨팅 도구) 및 정보자원’, ‘공유 및 평가’와 그 개념과 성격, 차원이 다르다. 셋째, 초기 개념 모형은 구성요소별 설계원리만 제시되어있기 때문에 수업 설계 시 고려해야 할 핵심적인 요소가 무엇인지 명확히 이해하기 어려우며, 이를 보완해 줄 수 있는 하위요소를 추가적으로 포함시킬 필요가 있다.

초기 개념모형은 위와 같은 전문가 의견을 반영하여 2차 개념모형으로 수정 및 도출되었다. 초기 개념모형의 수정사항은 다음과 같다.

첫째, 개념모형을 구성하는 구성요소를 변경하였다. 초기 개념모형을 이루는 구성요소 중 하나인 ‘안내’는 다른 구성요소보다 하위 단계의 개념이고, 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계의 핵심요소를 반영한다고 보기 어렵다는 전문가의 의견을 반영하여 삭제하였다. 또한 ‘협력적 학습 환경’도 연구의 특성을 드러내기 어려우며, 구성요소라기 보다는 일반적인 수준의 수업 설계 하위 요인에 가깝다는 전문가의 의견을 고려하여 삭제하였다.

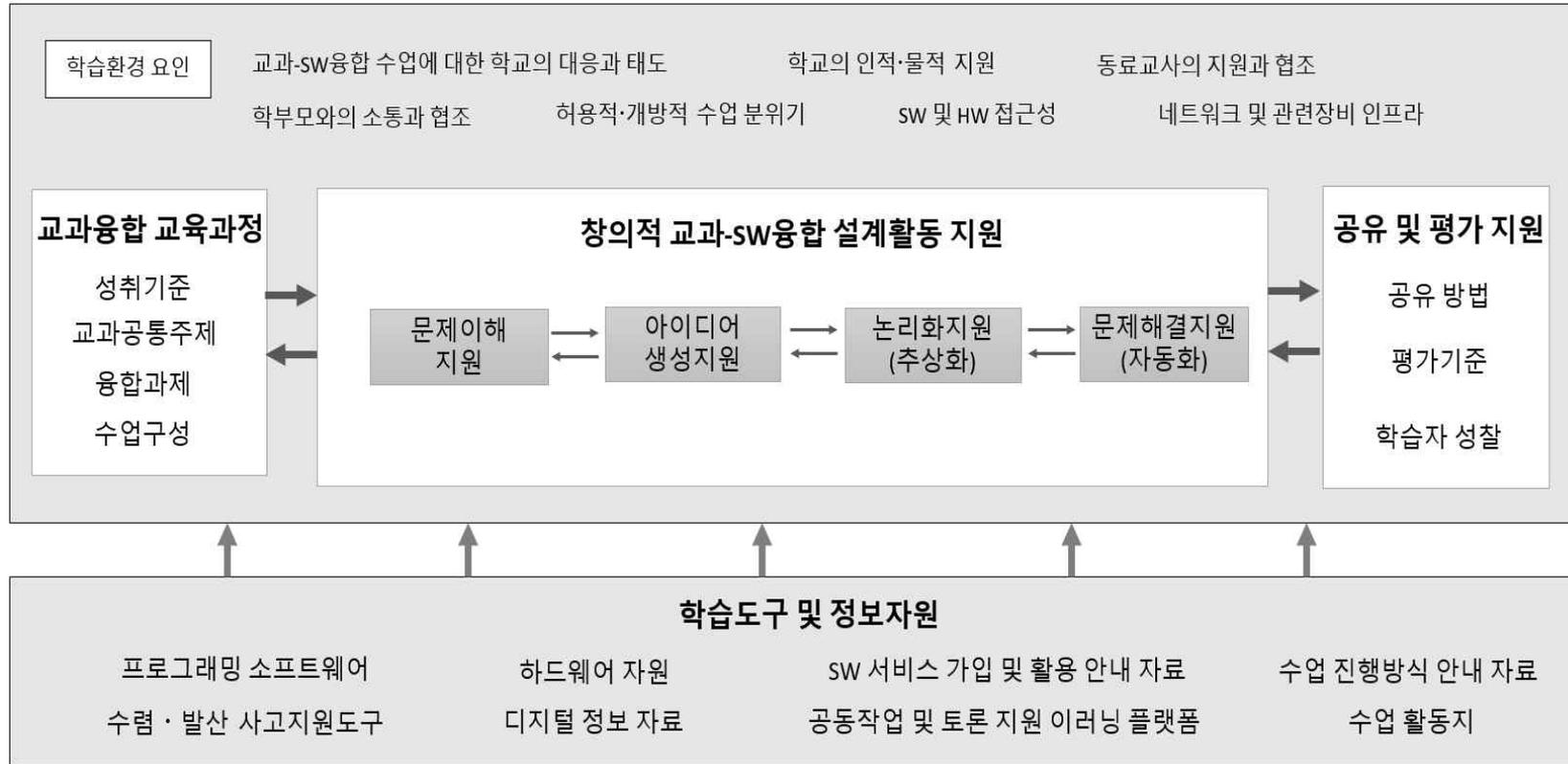
둘째, 구성요소의 명칭을 변경하고 하위요소를 도출하였다. 구성요소의 개념과 설명을 명확히 제시하고, 핵심적으로 고려해야 할 사항을 전달할 수 있도록 하위요소를 도출하여 개념모형에 포함하였다. 더불어 교과-소프트웨어융합 수업이라는 특성을 포괄할 수 있도록 초기 모형의 ‘컴퓨팅 사고력(CT)기반 창의적 융합설계활동 설계’를 ‘창의적 교과-소프트웨어융합 설계활동 지원’으로 수정하였으며, 구성요소의 주체를 명확히 드러내기 위해 ‘공유 및 평가’를 ‘공유 및 평가 지원’으로 수정하였다. 또한 초기 모형의 구성요소인 ‘교과융합 문제’는 전반적인 융합수업의 재구성이라는 의미를 포괄하지 못하는 개념으로 판단하여 ‘교과융합

교육과정'으로 구성요소의 명칭을 변경하였다.

셋째, 학습 환경 요인을 추가하였다. 학습 환경 요인은 교과-소프트웨어융합 수업 설계를 수행하기 위해 갖추어야 할 요소들로서, 학교의 대응과 태도, 학교의 인적/물적 지원, 동료교사의 지원과 협조, 학부모와의 소통과 협조, 허용적/개방적 분위기, 소프트웨어 및 HW 접근성, 네트워크 및 관련 장비 인프라 등 학습 활동 설계에 영향을 미치는 다양한 환경 변수들이다.

넷째, 개별 구성요소 간의 관계를 표현하기 위해 구성요소의 위치를 조정하고 화살표를 사용하여 도식화 하였다. 수정된 2차 개념모형은 '창의적 교과-소프트웨어융합 설계활동 지원'을 중심으로 이루어진다. 이러한 설계활동의 전후로 '교과융합 교육과정' 구성을 통해 전반적인 활동의 방향성을 설정하고, '공유 및 평가 지원'을 통해 전체 활동을 마무리한다. 세 구성요소는 학습 환경 요인에 의해 영향을 받는다. 학습 환경 요인에 영향을 받는 '교과융합 교육과정', '창의적 교과-소프트웨어융합 설계활동 지원', '공유 및 평가 지원'의 세 구성 요소는 '학습도구 및 정보자원'의 지원을 받게 된다. 수정된 2차 개념모형은 이러한 구성요소 간의 상호작용을 박스와 화살표를 활용하여 시각적으로 표현하였다.

위와 같은 수정사항을 반영하여 도출된 2차 개념모형은 [그림 IV-3]와 같다.



[그림 IV-3] 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계의 2차 개념 모형

## 2) 2차 전문가 타당화 결과

2차 개념모형의 타당성, 설명력, 유용성, 보편성, 이해도에 대한 전문가 타당화 점수는 평균 3.40점에서 3.80점 사이에 분포하였다(<표 IV-17> 참조). CVI는 모든 항목에서 1.00으로 나타나 전문가들은 2차 개념모형이 타당하다고 평가하였다는 것을 알 수 있다. 또한 전문가들의 평가를 신뢰할 수 있는지를 알아보는 IRA도 1.00으로, 2차 개념모형에 대한 전문가 타당화 결과가 신뢰할 수 있는 것으로 나타난다.

<표 IV-17> 2차 개념모형에 대한 2차 전문가 타당화 결과

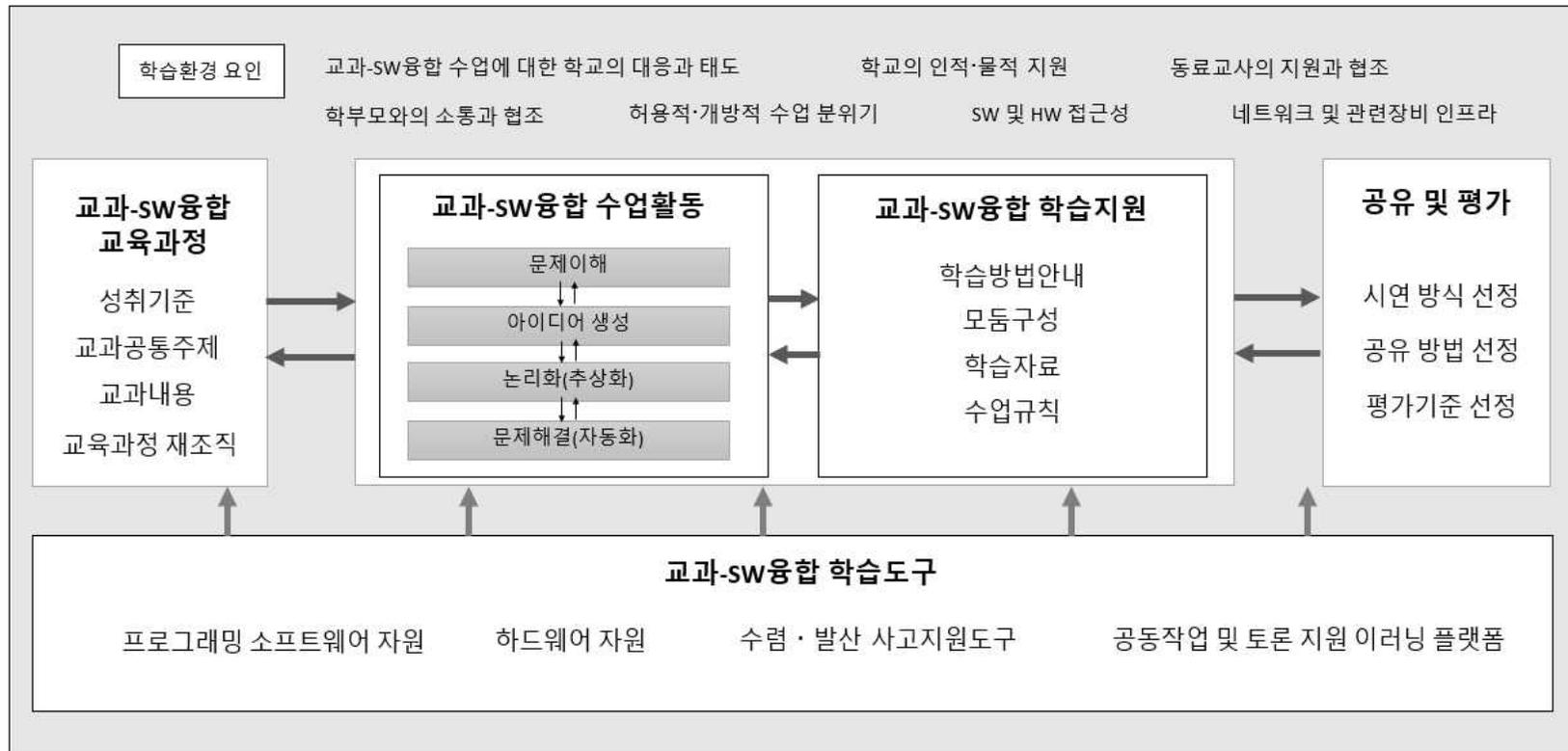
영역	전문가					평균	표준 편차	CVI	IRA
	B	C	F	G	H				
타당성	3	3	4	4	3	3.40	.548	1.00	
설명력	3	4	4	4	4	3.80	.447	1.00	
유용성	3	4	4	4	4	3.80	.447	1.00	1.00
보편성	3	4	4	4	4	3.80	.447	1.00	
이해도	3	4	4	4	3	3.60	.548	1.00	

2차 개념모형에 대해 전문가들이 제안한 의견은 다음과 같다. 첫째, ‘창의적 교과-소프트웨어융합 설계활동 지원’ 구성요소는 수업 활동 내용을 주로 포함하고 있어 구체적으로 수업활동을 어떻게 지원할 수 있는지를 알 수 있도록 추가적인 구성요소가 필요하다. 둘째, 일부 하위 구성요소의 위계와 내용이 적절치 않아 수정이 필요하다. 이와 같은 전문가들의 의견을 반영하여 다음과 같은 수정을 통해 3차 개념모형을 도출하였다.

첫째, ‘창의적 교과-소프트웨어융합 설계활동 지원’ 구성요소를 ‘교과-소프트웨어융합 수업활동’과 ‘교과-소프트웨어융합 학습지원’의 두 가지 구성요소로 구분하였다. ‘교과-소프트웨어융합 학습지원’은 ‘교과-소프트

웨어융합 수업활동'을 지원하는 구성요소로서 학습방법 및 수업규칙 안내, 모듈 구성, 학습 자료를 하위 요소로 포함한다. 둘째, 상위 구성요소에 적합한 내용으로 하위 요소들을 수정, 추가, 삭제하였다. 예컨대 '교과-소프트웨어융합 학습도구'의 하위요소를 일부 변경하였는데, 이는 수업 전반을 이러닝 플랫폼을 통해 제공하는 것으로 가정했을 때 일부 수업 활동지, 수업 진행방식 안내 자료, 디지털 정보 자료 등은 플랫폼과 연계하여 제공될 수 있으며, 이에 따라 하위 요소가 조금 더 단순한 형태로 제시될 필요가 있다는 전문가 의견을 반영하여 하위 요소에서 삭제했기 때문이다. 셋째, '교과-소프트웨어융합 학습도구'를 학습 환경 요인에 포함되도록 모형의 도식을 변경하였다. '교과-소프트웨어융합 학습도구'는 프로그래밍 소프트웨어 자원, 하드웨어 자원, 수렴·발산 사고지원 도구, 공동작업 및 토론 지원 이러닝 플랫폼을 하위요소로서 포함하고 있는 구성요소이며, 이는 학교의 인적, 물적 자원 및 장비 인프라 등을 포괄하는 학습 환경 요인을 바탕으로 제공될 수 있기 때문에 개념 모형의 도식을 변경하였다.

위와 같은 수정사항을 반영하여 도출된 3차 개념모형은 [그림 IV-4]와 같다.



[그림 IV-4] 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계의 3차 개념 모형

## 라. 절차 모형에 관한 전문가 타당화

### 1) 1차 전문가 타당화 결과

절차 모형에 대하여 모형의 타당성, 설명력, 유용성, 보편성, 이해도 측면에서 검토하는 전문가 타당화를 진행하였다. 초기 절차 모형에 대한 1차 전문가 타당화 검사 결과는 다음 <표 IV-18>와 같다.

<표 IV-18> 초기 절차모형에 대한 1차 전문가 타당화 결과

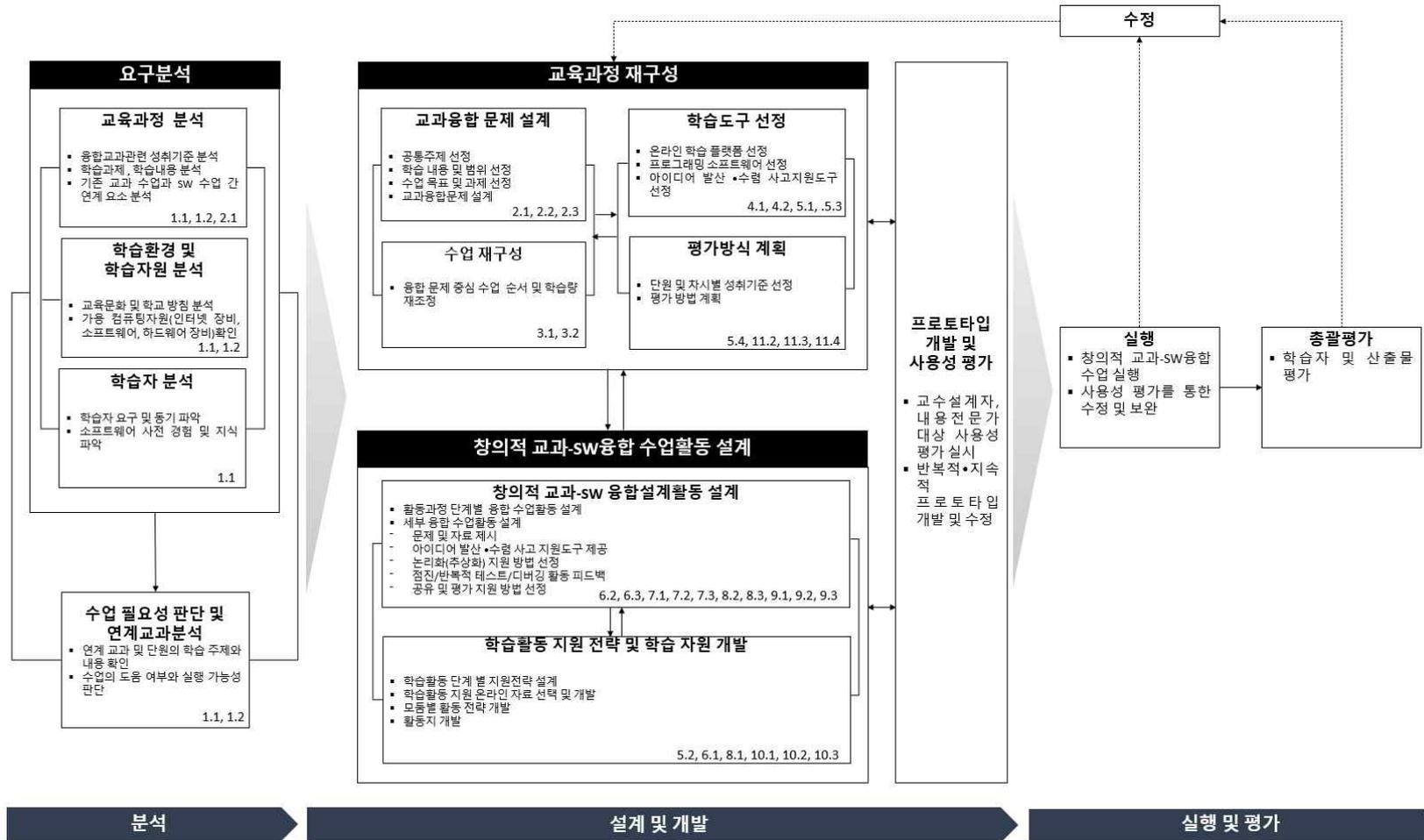
영역	전문가					평균	표준 편차	CVI	IRA
	A	B	C	D	E				
타당성	2	2	3	1	2	2.00	.707	0.20	
설명력	3	2	2	2	2	2.20	.447	0.20	
유용성	2	2	3	1	3	2.20	.837	0.40	0.00
보편성	2	2	3	2	2	2.20	.447	0.20	
이해도	3	2	3	2	2	2.40	.548	0.20	

1차 전문가 타당화 검사 결과 평균은 2.00점~2.40점 사이에 분포하는 것으로 나타났으며, 내용타당도 지수인 CVI는 0.20~0.40 사이로 나타나 초기 개념모형은 전체적으로 타당하지 않은 것으로 해석할 수 있다. 또한 평가자간 일치도인 IRA 점수는 0.00으로 평가자 간 일치도가 낮게 나타났다. 이는 평가자 D가 1점~2점으로 평가한 반면, 평가자 C는 2점~3점으로 평가하여 IRA가 낮아진 것으로 볼 수 있다. 이와 같은 1차 전문가 타당화 결과는 초기 절차모형은 전반적인 수정과 보완이 필요하다는 것으로 해석할 수 있다.

초기 절차 모형에 대한 전문가들의 의견을 종합하면 다음과 같다. 첫째, 개발된 원리가 각 수업모형의 어느 단계에 활용되는 것인지 매칭이 필요하다. 둘째, 분석이 너무 많은 단계로 이루어져 있어 단계의 수정 및

통합이 필요하다. 셋째, 교과-소프트웨어융합 수업이 필요한지, 그리고 연계할 수 있는 교과는 무엇인지를 파악하는 단계가 새롭게 추가될 필요가 있다. 넷째, 전반적으로 절차모형의 이미지를 정리하고 단순화할 필요가 있다.

초기 절차모형은 위와 같은 전문가 의견을 반영하여 2차 절차모형으로 수정 및 도출되었다. 초기 절차모형의 수정사항은 다음과 같다. 첫째, 수업설계 원리와 세부 지침을 절차모형의 각 단계에 제시하여 실제 수업설계 시 참고할 수 있도록 수정하였다. 둘째, 요구 분석 단계의 일부 하위 단계를 통합 및 삭제하였다. ‘교육문화 분석’과 ‘학습자원 분석’을 ‘학습 환경 및 학습자원 분석’으로 통합하였으며, 실질적으로 학교 외부의 전문가 그룹을 섭외하여 수업 설계를 할 수 있는 기회는 많지 않다는 전문가의 의견을 반영하여 ‘전문가그룹 분석’을 삭제하였다. 셋째, 요구분석 단계 이후 ‘수업 필요성 판단 및 연계교과 분석’ 단계를 추가하였다. 넷째, 프로토타입 개발 및 사용성 평가를 설계 및 개발 단계의 주요 단계로 인식할 수 있도록 위치를 조정하였으며, ‘교육과정 재구성’과 ‘창의적 교과-소프트웨어융합 수업활동 설계’ 단계와 지속적인 상호작용 관계가 나타나도록 상호 방향의 화살표를 표기하였다. 위와 같은 수정사항을 반영하여 도출된 2차 절차모형은 [그림 IV-5]와 같다.



[그림 IV-5] 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계의 2차 절차 모형

## 2) 2차 전문가 타당화 결과

2차 모형에 대한 2차 전문가 타당화 결과는 평균 3.60점에서 3.80점 사이에 분포하여 비교적 높은 점수를 받았으며, CVI도 모든 항목에서 1.00을 나타내 절차모형에 대한 전문가 타당화 검사 결과는 타당하다고 해석할 수 있다. 또한 IRA도 1.00으로 전문가 의견의 일치도가 높은 것으로 나타나 검사 결과가 신뢰할 만한 것으로 판단할 수 있다.

<표 IV-19> 2차 절차모형에 대한 2차 전문가 타당화 결과

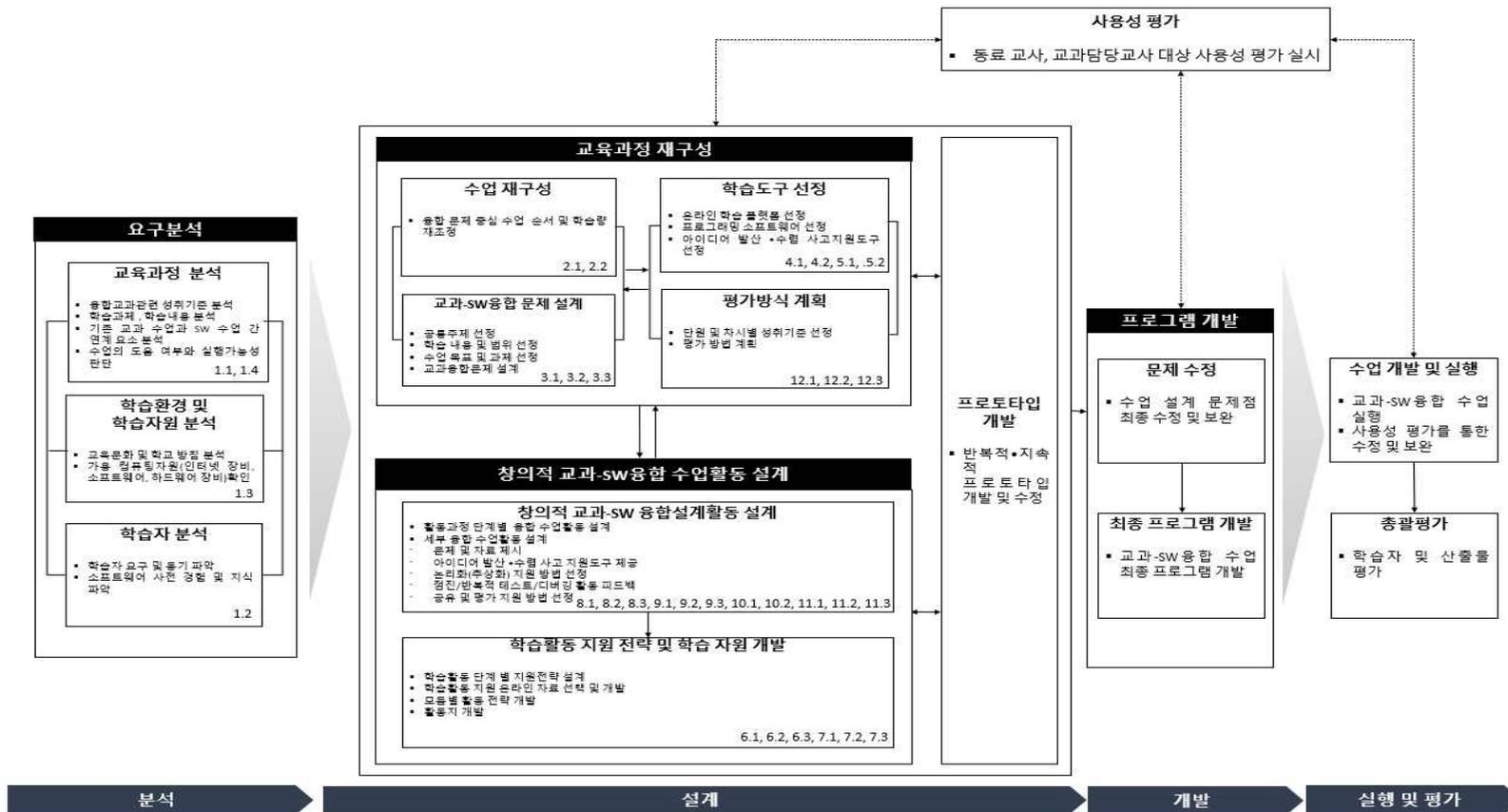
영역	전문가					평균	표준 편차	CVI	IRA
	B	C	F	G	H				
타당성	3	4	4	4	4	3.80	.447	1.00	
설명력	3	3	4	4	4	3.60	.548	1.00	
유용성	3	4	4	4	4	3.80	.447	1.00	1.00
보편성	3	4	4	4	4	3.80	.447	1.00	
이해도	3	4	4	4	4	3.80	.447	1.00	

절차모형에 대한 전문가 타당화 의견을 정리해보면 다음과 같다. 첫째, 수업의 설계 단계 이후 프로그램 개발 단계를 추가할 필요가 있다. 둘째, 사용성 평가는 설계, 개발, 실행 단계를 아울러 이루어지며 이에 대한 수정 및 보완이 이루어져야 한다. 셋째, 수업 필요성 판단 및 연계 교과 분석 단계는 요구분석의 하위 단계로 보여 위치를 조정할 필요가 있다. 2차 절차모형은 위와 같은 전문가 의견을 반영하여 3차 절차모형으로 수정 및 도출되었다. 2차 절차모형의 수정사항은 다음과 같다.

첫째, 수업 설계 및 프로토타입 개발 이후 설계한 수업의 문제점을 수정, 보완하고 하고 최종 프로그램을 개발하는 프로그램 개발 단계를 추가하였다. 둘째, 프로토타입 개발 및 사용성 평가 단계를 분리하였다. 먼저 설계 단계에서는 교육과정 재구성을 거쳐 창의적 교과-소프트웨어융합 수업활동 설계에 대한 프로토타입을 반복적, 지속적으로 개발하여 수

정하는 과정이 이루어지며, 사용성 평가는 설계단계와 프로그램 개발 단계, 그리고 수업 설계 단계에서 동료 교사 혹은 교과담당교사를 대상으로 지속적으로 실시하여 수정 및 보완을 하는 과정으로 수정되었다. 셋째, 수업 필요성 판단 및 연계교과 분석 단계를 교육과정 분석 단계의 하위 단계로 위치를 이동하여 수정하였다.

2차 절차모형은 위와 같은 전문가 의견을 반영하여 3차 절차모형으로 수정 및 도출되었다([그림 IV-6] 참고).



[그림 IV-6] 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계의 3차 절차 모형

## 마. 사용성 평가

개발된 개념모형, 절차모형, 설계원리가 실제 학교 현장에서 수업을 설계하는 교사들이 사용하기에 적합한지 검토하기 위하여 사용성 평가를 실시하였다. 사용성 평가에는 서울과 대구, 충남에 위치한 초등학교에 근무하고 있는 초등교사 3인이 참여하였다. 사용성 평가에 참여한 세 명의 교사는 개발된 모형과 원리에 대한 안내를 듣고 이해한 후 자신이 직접 교과-소프트웨어융합 수업을 진행한다는 가정 하에 교수·학습 설계안을 작성하였다([그림 IV-7] 참조).

교과 명	국가	대 상	요동 세션	요동비율	그것이 알고 싶다: 크로니카	주 제	한정문제를 일컫는 정보기법(연표) 만들기	차 시	3/8	수업 모형
단 단 명	1. 비유하는 표현				크로니카 사례로 사회가 일어나는 여러 현상들을 하나의 큰 흐름적으로 설명하고 관련 주제 고교제를 완성 중심의 체험이 일어나는 학습과정과 연결하여 참여와 인식 확장 을 신장시키고자 함	학습 목표	정보기법, 제작을 위해 도출한 아이디어를 토대로 알고리즘을 설계할 수 있다.			
차 시	2. 비유하는 표현					자 료	교 시   교과용 컴퓨터 학 생   크롬북, 활동지, 배움넷			
지도 교사	사유기			지도 장소	수-3					
학습 목표	1. 비유하는 표현 및 비유하는 표현을 설명할 수 있다. 2. 비유하는 표현을 설명할 수 있다. 3. 비유하는 표현을 설명할 수 있다.				2020년 크로니카 사례는 우리 생활에 거대한 영향을 미쳤다. 크로니카 가장 간단한 일은 장례의 관한 564년 고과관계에 크로니카의 관련자를 수 있는 성취기준으로 프로그램 학습으로 학습자에게 설명하고 거대한 주제로 연결할 수 있으며, 크로니카를 수렴하는 거대한 참여와 인식을 요구적으로 기술 수 있다.					
수업 모형	교수학습활동	수업 형태	자료 및 도구	활동구분	관련 고교2차 주제	교과 학습내용	원시자료 요소			
사전 학습	비유하는 표현 찾기 - 활동과 비유 시에서 비유하는 표현을 찾아서 문맥 에 답해본다.	안내	주말 홈즈	크로니카 뉴스	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	뉴스에 나타난 정보의 특성 및 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	최근 주제를 따른 스토리보드 작성 하기 기사를 읽고리즘을 설계하고 구현하기			
주제발문	비유하는 표현에 대한 소개하기 - 비유하는 표현에 대한 설명을 설명해본다. - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	영상 시청 발표	*무빙	크로니카 뉴스	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.			
포럼 (5)	학습목표 제시하기 - 학습목표와 함께 발표하는 표현을 설명할 수 있다.	안내	PPPT	크로니카 뉴스	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.			
학습안내	학습 안내하기 - 학습안내의 학습할 내용을 안내한다.	안내	PPPT	크로니카 뉴스	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.			
평가	평가 관련 지시 주재 찾기 - 평가 관련 지시 주재 찾기 - 평가 관련 지시 주재 찾기 - 평가 관련 지시 주재 찾기	안내	PPPT	크로니카 뉴스	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.			
평가	평가 관련 지시 주재 찾기 - 평가 관련 지시 주재 찾기 - 평가 관련 지시 주재 찾기 - 평가 관련 지시 주재 찾기	안내	PPPT	크로니카 뉴스	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.	비유하는 표현과 비유하는 표현의 차이 - 비유하는 표현의 사용 목적을 설명할 수 있다. - 비유하는 표현의 사용 목적을 설명할 수 있다.			

[그림 IV-7] 수업 모형과 원리를 바탕으로 작성한 교수·학습 설계안

사용성 평가에 참여한 세 명의 교사는 교수·학습 설계안을 작성한 이 후 제공된 사용성 평가 질문지에 따라 모형과 원리의 사용성을 평가해 주었다. 다음은 개념모형, 절차모형, 설계원리를 바탕으로 수업을 설계했 을 때 도움이 되었는지를 묻는 사용성 평가 문항에 대한 결과이다(<표 IV-20>).

<표 IV-20> 개념모형, 절차모형, 설계원리에 대한 사용성 평가 결과

사용성 평가 문항	평균	표준 편차	CVI	IRA
개념모형과 그 설명은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업설계가 무엇인지 이해하는데 도움이 된다.	4.00	.000	1.00	1.00
개념모형과 그 설명은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업설계를 위하여 무엇을 해야 하는지(어떤 요소들을 고려해야 하는지) 이해하는데 도움이 된다.	3.67	.078	1.00	
절차모형과 그 설명은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업을 설계하기 위해 어떠한 단계를 따라야 하는지를 잘 설명하고 있다.	3.67	.078	1.00	
절차모형과 그 설명은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업을 설계하는데 도움이 된다.	3.67	.078	1.00	
설계원리 및 상세지침은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업을 설계하는데 도움이 된다.	4.00	.000	1.00	

개념모형과 그 설명은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업설계가 무엇인지 이해하는데 도움이 되었는지에 대한 문항에 대해서 세 명의 교사 모두 ‘매우 그렇다(4점)’고 응답하였다. 개념모형과 그 설명이 수업 설계를 위해 무엇을 해야 하는지 이해하는데 도움이 되었는지에 대한 문항에 대해서는 평균 점수가 3.67점, 절차모형에 문항의 평균 점수도 3.67점으로 나타나 전반적으로 도움이 되었으나, 상대적으로 개선의 여지가 있는 것으로 나타났다. 설계 원리 및 상세지침이 수업을 설계하는데 도움이 되었는지에 대한 문항에 대해서는 교사 세 명 모두 ‘매우 그렇다(4점)’으로 답하였다.

다음 <표 IV-21>는 사용성 평가의 주관식 개방형 문항에 제시된 의견을 정리한 것이다.

<표 IV-21> 사용성 평가에서 제안된 개념모형, 절차모형, 설계원리의 강점, 약점, 개선점 및 추가의견

	강점	약점	개선점
개념 모형	<ul style="list-style-type: none"> <li>수업 설계 과정에서 놓칠 수 있는 요소들을 모두 확인할 수 있음</li> <li>각 유목들의 위치가 직관적이어서 중심이 되는 요소와 지원해 주는 요소 간 구별이 명확함</li> <li>‘공유 및 평가 지원’ 단계가 있어 수업과 평가의 일체화를 가능케 하는 요소가 되었음</li> </ul>	<ul style="list-style-type: none"> <li>학습 환경 요인 부분에서 유목화가 이루어질 필요가 있어 보임</li> <li>‘공유 및 평가’는 이미 평가기준이 선정된 상태에서 실제 평가가 이루어지는 단계이므로, 평가기준 선정을 교과-소프트웨어융합교육과정 단계로 이동시키는 것을 제안함</li> </ul>	<ul style="list-style-type: none"> <li>예시를 보다 구체적으로 제시하는 것이 필요함</li> <li>일부 용어의 수정이 필요함(예: ‘프로그래밍 소프트웨어’를 초등교사에게 익숙한 ‘EPL’로 수정)</li> </ul>
절차 모형	<ul style="list-style-type: none"> <li>각 요소들이 직관적이고 잘 조직되어 있음</li> <li>실제 현장에서 사용하기에 적합함</li> <li>수업 설계 및 실행 후 보완을 통해 지속적인 개발이 가능함</li> </ul>	<ul style="list-style-type: none"> <li>프로토타입 개발과 프로그램 개발 간의 차이가 별개의 요소로 나누어질 만큼 클지 의문이 듦</li> <li>‘요구분석’ 단계가 시간과 노력이 많이 요구되는 경향이 있음</li> <li>‘수업 재구성’의 내용이 ‘교육과정 재구성’과 같음</li> </ul>	<ul style="list-style-type: none"> <li>프로토타입 개발 단계와 프로그램 개발 단계 간의 중첩성 해결이 필요함</li> <li>‘교육과정 재구성’과 ‘교과-소프트웨어융합 활동’의 유목이 일부 겹치는 것 같아 유목 조정이 필요함</li> </ul>

설계 원리	<ul style="list-style-type: none"> <li>• 설계원리가 자세한 예시와 함께 제시되어 있어 현장에서 쉽게 활용할 수 있을 것 같음</li> </ul>	<ul style="list-style-type: none"> <li>• 일부 예시(활용할 수 있는 온라인 소프트웨어 등)가 좀 더 구체적이면 좋겠음</li> <li>• 교육과정과 교과 소프트웨어 수업 설계 간의 관계에 대해서 고려할 필요가 있음</li> </ul>	<ul style="list-style-type: none"> <li>• 현장 교사에게 익숙한 용어로 수정이 필요함</li> </ul>
추가 사항	<ul style="list-style-type: none"> <li>• 학교의 방침과 문화를 파악하는 설계원리의 경우에 구체적인 해결방법이 명시되지 않음. 파악 이후의 단계에 대한 세부적인 지침이 있다면 현장에서 도움이 될 것 같음</li> <li>• ‘점진적 문제해결 지원성의 원리’와 관련하여 ‘문제적 상황에 처할 학습자에게 제공되는 피드백’을 어떻게 제공해야 하는지 설명을 해준다면 도움이 될 것 같음. 학습자에게 생길 변수나 학습자가 실패에서 포기하지 않고 계속해서 문제 상황에 도전할 수 있도록 하는 기반을 교수자로서 어떻게 제공하여야 하는지에 대한 구체적인 설명이 추가 되면 좋겠음</li> </ul>		

사용성 평가 질문지를 통해 세 명의 현장 교사에게 개념 모형, 절차 모형, 설계 원리에 대한 강점, 약점, 개선점 및 추가가 필요한 사항에 대한 답변 결과는 다음과 같다.

개념 모형은 수업설계 시 필요한 요소들이 모두 포함되어 있어 수업 설계과정에서 놓칠 수 있는 요소들을 확인할 수 있다는 점이 대표적인 장점으로 꼽힌 반면, 구성요소의 하위 요소 및 학습 환경 요인의 재유목화가 필요하다는 점이 약점으로 지적되었다. 개선점으로는 구체적인 예시 제공 및 용어 수정에 대한 필요성이 제시되었다.

절차모형은 각 요소들이 직관적으로 조직되어 있으며, 래피드 프로토타이핑 방법론의 적용을 통해 지속적인 수업 개발 및 보완이 가능하다는 점이 장점이라는 의견이 있었다. 반면 일부 단계가 중첩성을 보여 유목을 조절하고 재구조화 할 필요가 있다는 개선점에 대한 의견도 제시되었

다. 또한 ‘수업 재구성’의 내용이 상위 단계인 ‘교육과정 재구성’의 의미와 일치하므로 ‘교육과정 재구성’으로 단계명을 수정할 필요가 있다는 의견이 제시되었다.

설계원리는 자세한 예시가 주어져 있어 실제 수업 설계 시 많은 도움이 되었다는 의견이 있었던 반면, 일부 예시가 구체적이지 않고, 현장 교사에게 익숙한 용어로 수정이 필요하다는 의견이 제시되었다.

이 밖에 추가적인 의견으로 학교의 방침과 문화를 파악하는 설계원리에 대해 구체적인 예시가 제공될 필요성이 있다는 의견과, 점진적 문제 해결 활동 시 학습자가 자신의 실패에 포기하지 않고 계속적으로 문제 상황에 도전할 수 있도록 어떻게 피드백을 제공해야 하는지에 대한 설명이 추가된다면 좀 더 쉽게 원리를 수업 설계에 적용시킬 수 있을 것 같다는 의견이 있었다.

개별 설계원리에 대한 사용성 평가 결과는 다음 <표 IV-22>와 같으며, 개별 원리 별 상세 지침에 대한 사용성 평가 결과는 [부록 7]에 제시하였다.

<표 IV-22> 개별 설계원리에 대한 사용성 평가 결과

설계원리	평균	표준 편차	CVI	IRA
1. 수업 실행성의 원리	3.67	.577	1.00	0.8
2. 맥락 재구성의 원리	4.00	.000	1.00	1.00
3. 실제적 문제의 원리	3.67	.577	1.00	1.00
4. 소프트웨어도구 적합성의 원리	4.00	.000	1.00	1.00
5. 사용 용이성의 원리	4.00	.000	1.00	1.00
6. 자료의 명료성의 원리	3.67	.577	1.00	1.00
7. 프로그램 모듈 구성의 원리	3.67	.577	1.00	1.00
8. 문제이해 지원성의 원리	3.67	.577	1.00	1.00
9. 아이디어 생성 지원성의 원리	4.00	.000	1.00	1.00
10. 논리화 지원성의 원리	4.00	.000	1.00	1.00

11. 점진적 문제해결 지원성의 원리	3.67	.577	1.00	1.00
12. 공유 및 평가 지원성의 원리	3.67	.577	1.00	1.00

전체 설계원리와 상세지침에 대한 사용성 평가 결과 평균 3.3점에서 평균 4.0점 까지 분포하고 있는 것으로 나타나 전체 설계원리와 상세지침은 전반적으로 양호한 수준임을 알 수 있다. 그러나 상세 지침 중 일부 문항에 대해서는 ‘그렇지 않다(2점)’는 응답을 확인할 수 있었다. 예를 들어 ‘1.4 프로그래밍 소프트웨어를 활용한 교과융합 수업이 성취 기준 달성에 도움이 되는지를 확인하라’에 대해서 현장교사는 교과-소프트웨어융합 수업은 성취기준 달성에 목적이 있는 것이 아니라, 최소 성취기준을 달성하면서 교육과정보다 더 많은 것을 지도하기 위해 이루어진다고 생각하므로 ‘타당하지 않다.’라는 의견을 주었다. 또한 ‘11.1 학습자가 점진적으로 프로그램을 테스트하고 디버깅할 수 있도록 피드백을 제공하라’에 대해서 타당하지 않다고 답한 현장교사는 학습자에게 생긴 변수나 학습자가 실패에서 포기하지 않고 계속 점진적으로 문제 상황에 도전할 수 있도록 하는 기반을 교수자로서 어떻게 제공해야 하는지 고민이 많은 부분이며, 이 부분의 설계하는 데 대하여 구체적인 예시나 설명이 필요하다는 의견을 주었다. 따라서 최종 원리 수정 시 이상의 상세지침 1.4, 상세지침 11.1에 대한 의견을 반영할 필요가 있다고 판단된다.

각 평가 문항에 대하여 CVI는 1.0이상, IRA는 0.8 이상인 것으로 나타나 설계원리 및 상세지침에 대한 현장교사의 사용성 평가 결과는 일치도가 높은 수준이며 신뢰할 수 있다는 것을 확인할 수 있다.

### 3. 외적 타당화

#### 가. 수업의 설계 및 실행

##### 1) 수업의 설계

수업설계 모형을 현장에 적용하고 평가하기 위해 연구자는 교수자와 세 차례의 사전 회의를 진행하며 모형과 원리를 설명하고 융합 교과 내용에 대해 협의하였다. 이 후 교수자는 개념 모형과 절차 모형, 설계 원리를 개별적으로 검토한 후, 이를 참고하여 총 6차시의 수업을 설계하였다(<표 IV-23> 참조). 교수자의 설계 과정은 다음과 같다. 우선 교수자는 연구자와의 사전 회의를 거치며 수업 실행의 원리와 맥락 재구성의 원리에 따라 소프트웨어와 연계할 교과와 단원, 학습 내용을 선정하고 수업 환경을 고려하여 수업의 실행 가능성과 효과성을 판단하고, 이를 하나의 맥락으로 재구성하는 설계 과정을 거쳤다. 교수자는 초등학교 6학년 2학기의 진도에 해당하는 다양한 과목과 단원을 구체적으로 분류하고 나열한 후, 교과간의 연계와 소프트웨어의 활용을 통해 효과적인 수업을 설계할 수 있을 것으로 판단되는 사회과의 환경문제 단원과 국어과의 발표활동 단원을 선정하였다. 또한 다양한 융합 수업 모듈을 연계하여 본 연구의 현장 평가 사전에 사회과목의 환경문제 단원을 바탕으로 학습자들이 환경문제를 알아보고 자신만의 생태도시를 만들어볼 수 있는 수업모듈을 실시하였다. 이를 통해 환경과 생태도시라는 큰 맥락을 이루는 모듈간의 연계가 이루어졌으며, 본 연구의 핵심 활동인 환경문제를 알리는 자신만의 생태도시 홍보자료 제작을 통해 코딩 소프트웨어를 활용하여 다양한 아이디어를 바탕으로 창의적인 산출물을 만들어보는 교과-소프트웨어융합 문제를 설계하였다. 교수자는 수업설계 원리로 제시되었던 소프트웨어 적합성의 원리 및 사용 용이성의 원리를 토대로 수업 목표를 달성하기에 적합하고 학습자들이 사용하기 용이한 블록코딩 소프트웨어인 엔트리를 선정하였고, 학습자들의 창의적 문제해결 활동을 지

원하기 위해 다양한 소프트웨어를 활용하여 수업 자료를 개발하였다. 또한 프로그래밍 모듈 구성의 원리에 따라 동료프로그래밍 활동이 가능하도록 다양한 수준의 학습자들로 모듈을 구성하였다. 즉, 교수자는 모형이 제시하는 핵심 요소와 개념, 원리와 절차를 바탕으로 활동을 구조화하고, 활용할 수 있는 소프트웨어를 선정하고, 이해를 돕기 위한 수업 자료를 수집하고 개발하였다.

한편 수업 설계 당시 코로나바이러스감염증-19의 영향으로 초등학교 수업이 전면적으로 온라인 수업의 형태로 진행되고 있어, 이를 고려하여 모든 차시를 온라인 수업으로 진행하는 일정으로 수업을 설계하였다.

<표 IV-23> 수업설계 원리 및 모형을 적용한 수업설계 개요

단원	사회 3단원-지속가능한 지구촌 국어 4단원-효과적으로 발표해요 실과 4단원-생활 속의 소프트웨어
주제	환경문제를 알리는 홍보자료(엔트리) 만들기
목표	일상생활에서 찾아볼 수 있는 환경문제를 해결하기 위한 노력을 하고 있는 다양한 생태도시를 찾아보고, 내가 계획한 생태도시의 홍보자료를 계획할 수 있다.
대상	초등학교 6학년
활용 소프트웨어	블록 코딩 소프트웨어(엔트리 : <a href="https://playentry.org">https://playentry.org</a> ) 패드렛( <a href="https://ko.padlet.com/">https://ko.padlet.com/</a> ) 구글독스( <a href="https://drive.google.com/">https://drive.google.com/</a> )
1차시	대도시 환경문제를 알아보고, 생태도시 필요성과 이를 알리기 위한 노력을 이해하기
2차시	홍보 애니메이션에 대한 아이디어 생각하기
3차시	아이디어를 선별하고 홍보 애니메이션 제작을 위한 논리적인 해결안 작성하기
4차시	블록 코딩 소프트웨어를 활용한 홍보 애니메이션 제작하기 1
5차시	블록 코딩 소프트웨어를 활용한 홍보 애니메이션 제작하기 2

## 2) 수업의 실행

본 수업을 시작하기 일주일 전부터 정규 수업을 통해 학습자들에게 프로그래밍 소프트웨어와 블록 코딩을 활용한 코딩 방법에 대해 기초적인 지식을 제공하였다. 또한 수업 전 학습자 스스로 생태도시를 설계하는 사회과의 환경 단원 수업을 실시한 후, 본 수업에서는 직접 설계한 환경도시의 홍보 애니메이션을 제작하는 흐름으로 구성하였다. 본격적인 수업은 학기 말인 12월 넷째 주부터 시작하였다. 연구의 취지를 설명한 후 학습자의 동의를 받기 위해 학부모 및 학습자에 대한 동의서를 배포하였으며, 일주일 후 수집하였다. 연구는 동의서를 수집한 다음 주부터 시작하였다.

6차시 수업의 전체적인 구조를 살펴보면, 생태도시 홍보의 필요성을 알아보고, 정보를 수집한 후 아이디어를 생성하고 스토리보드를 작성하며 구조화시킨다. 이를 토대로 코딩 소프트웨어를 활용하여 홍보 애니메이션을 만든 후, 모듈별로 산출물을 온라인 플랫폼에 탑재하고 학습자들 간에 평가가 이루어지도록 하였다.

한편 본 수업의 실행 직전 코로나바이러스감염증-19로 인한 사회적 거리두기가 3단계로 격상되었다. 이러한 이유로 오프라인 수업이 불가능하게 되어 모든 수업이 온라인 상에서 화상회의 소프트웨어를 활용하여 진행되었다. 교수자는 수업 동안 학생들의 비디오를 모두 on 상태로 유지하도록 안내하였으며, 모듈 별 활동을 위해 줌(ZOOM)의 소회의실 기능을 활용하여 예정했던 인원대로 그룹을 나누었다. 모듈활동 시에는 교수자가 각 그룹 소회의실에 입장하여 학습자들의 활동 진행 상황을 점검하는 시간을 가졌다. 또한 일정 상 수업을 나홀에 걸쳐 1~2차시, 3차시, 4~5차시, 6차시로 모듈화하여 실시하게 되었다.

가) 1차시~2차시 : 대도시 환경문제를 알아보고, 생태도시 필요성과 이를 알리기 위한 노력을 이해하기(문제정의 및 아이디어 생성)

1차시~2차시 수업에서는 환경 문제와 환경 도시를 알아보고, 생태도시 홍보의 필요성을 제시한 후, 홍보 애니메이션 제작을 위한 아이디어를 생성하는 수업이 진행되었다. 우선 사전에 생태도시를 다룬 도서를 미리 읽어오도록 안내하고, 수업 시작 시 교수자가 대도시 환경오염 문제의 심각성을 다룬 영상을 제작하여 줌(ZOOM)의 화면 공유를 통해 상영하였다. 이후 학생들은 환경오염을 극복하고 예방하기 위해 설계된 생태도시의 중요성과 이를 알리기 위한 홍보의 노력에 대해 생각해 보는 시간을 가졌다. 교수자는 학생들에게 질문을 하였고, 의견이 있는 학생들은 돌아가며 답변하는 형식으로 진행되었다. 이후 프로그래밍 소프트웨어 사이트(엔트리) 상에 업로드된 홍보 애니메이션 및 게임 등을 시연하며 최종적인 산출물의 형태를 예시로 제시하였다.

수업 이전 사회과 수업에서 계획한 자신만의 생태도시가 실제 다른 생태도시들과 비교했을 때 어떠한 특징을 가지는지를 비교하기 위해 국내외의 다양한 생태도시 자료를 수집, 분석한 후 공통점과 차이점을 정리하여 패들렛에 작성하도록 하였다.

이후 공통점과 차이점을 바탕으로 자신만의 생태도시 홍보를 위한 홍보 애니메이션의 컨셉 아이디어를 자유롭게 작성하여 패들렛에 업로드하고 공유하도록 하였다. 교수자는 공통점과 차이점을 작성한 이유에 대해 질문하였고, 이를 통해 도출한 컨셉 아이디어에 대해 피드백 하였다. 학습자는 피드백을 반영하여 새로운 카드를 패들렛에 추가하였다. 특히 교수는 학습자가 기본적인 애니메이션의 토대를 구축할 수 있도록 시간의 흐름과 장면 및 대사에 대한 아이디어를 추가하도록 피드백 하였고, 학습자들은 이를 반영하여 자유롭게 아이디어를 도출하고 카드를 작성하였다.

마지막으로 교수는 다른 학습자들의 의견 카드에 ‘마음’을 표시하고 좋은 점, 나쁜점, 흥미로운 점을 나누어 댓글을 추가하도록 하였다. 학습

자들은 공감하거나 좋다고 생각되는 아이디어에 ‘마음’을 누르고, 서로의 댓글에 피드백을 주었다. 학습자는 피드백을 정리하여 자신의 아이디어를 수정하고, 가장 많은 ‘마음’을 얻은 아이디어를 최종 아이디어로 결정하도록 하였다.



[그림 IV-8] 아이디어 생성을 위한 패들렛 화면

나) 3차시 : 아이디어를 선별하고 홍보 애니메이션 제작을 위한 논리적인 해결안 작성하기(스토리보드 작성 및 알고리즘 작성)

3차시 수업에서는 이전 차시에서 제작한 아이디어를 정리한 후 이를 토대로 홍보자료의 내용을 구성하고, 스토리보드를 작성하는 수업이 진행되었다. 수업 전반에는 이전 차시에서 패들렛 상에 작성한 모듈별 아이디어를 검토하며 우리 모듈의 생태도시와 다른 생태도시와의 공통점, 차이점, 홍보 애니메이션의 컨셉 및 아이디어를 다시 한 번 공유하고 정리하는 시간을 가지도록 하였다. 이후 ‘라면 끓여먹기’와 같은 실생활 속의 현상과 경험을 알고리즘의 순서도 형태로 표현한 자료를 제공하며 스토리보드를 바탕으로 알고리즘을 작성하는 예시를 제시하였다.

스토리보드는 배경과 등장 오브젝트, 시간을 장면 별로 설명하며 작성할 수 있는 양식을 사전에 제공하였으며, 교사는 구글독스에 동일한 양식을 작성하고 모듈별로 문서를 생성하여 링크를 공유하였다.

학습자들은 줌(ZOOM)의 소회의실 단위로 소그룹을 이루어 함께 협력하며 구글독스 상에서 스토리보드를 작성하였다. 교수자는 각 모둠의 소회의실에 참여하여 진행상황을 검토하고 스토리보드 작성에 대한 피드백을 제공하였다. 이후 모둠별 활동을 끝내고 스토리보드로 작성한 내용을 구현하기 위해서 어떤 순서대로 알고리즘을 짜면 좋을지에 대해 예시를 제공하였다. 교수자는 학습자들의 스토리보드에 문장 형태로 작성한 장면 설명을 바탕으로 절차를 나누어 순서도의 형태로 표현하는 일련의 과정을 구체적으로 설명하였다. 수업 후반에는 일부 모둠의 스토리보드에 대한 발표를 진행하였고 이에 대한 교수자의 피드백이 이루어졌다. 교수자는 온라인의 학급 게시판에 모둠별 주제와 내용을 정리하여 구글독스의 주소와 함께 업로드 하도록 안내하였다.



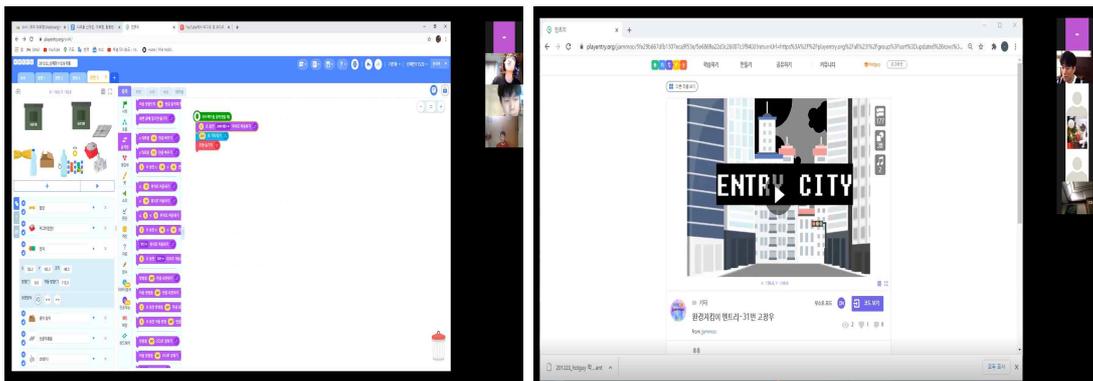
[그림 IV-9] 모둠별 스토리보드 작성 장면 및 스토리보드 산출물

**다) 4~5차시 : 환경문제를 알리는 홍보자료를 프로그래밍 소프트웨어를 활용하여 제작하기(프로그래밍을 통한 문제해결)**

4~5차시 수업에서는 환경문제를 알리는 홍보 애니메이션을 프로그래밍 소프트웨어를 활용하여 제작하도록 하였다. 학습자들은 지난 차시에 작성한 스토리보드와 알고리즘을 활용하여 블록형 프로그래밍 소프트웨어를 활용하여 애니메이션을 제작하였다. 본격적인 제작에 앞서 교수자는 사전에 학습한 프로그래밍 소프트웨어(엔트리)를 간단히 복습할 수

있도록 핵심적인 개념과 제작 방법을 설명하고, 이미지 및 오브젝트 등의 자료 검색 방법 및 엔트리 상에 탑재되어있는 음원을 삽입하는 방법을 안내 및 시연 하였다.

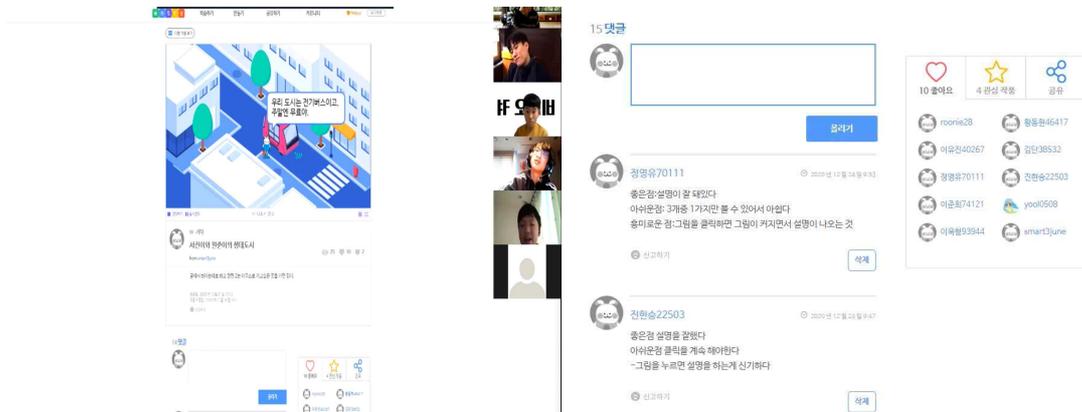
이후 학습자들이 모듈별로 소회의실에서 학습활동이 진행될 수 있도록 소그룹을 나누어 진행하였다. 학습자들이 애니메이션 제작을 위해 필요한 이미지, 오브젝트, 음원 등의 자료를 수집할 수 있는 시간을 제공하고, 이를 활용하여 본격적으로 홍보자료를 제작할 수 있도록 안내하였다. 이 때 모듈원 중 한명이 엔트리 화면을 공유하며 코드를 작성 및 입력하는 역할을 담당하였고, 다른 모듈원들은 미리 작성한 스토리보드와 알고리즘을 참고하며 계획한 장면대로 작성되고 있는지, 작성되고 있는 코드가 논리적이고 오류가 없는지를 확인하는 역할을 담당하였다. 또한 프로그래밍 작업 중 의도한 대로 작성되고 있는지를 알아보기 위해 반복적, 지속적으로 산출물을 실행하며 코드의 오류와 장면 연출을 수정할 수 있도록 하였다. 필요 시 엔트리 사이트에 업로드 되어있는 다른 사용자의 코드를 참고하여 작업할 수 있도록 안내하였다. 수업 후반에는 모듈별로 제작한 생태도시 홍보자료를 엔트리 사이트에 개설된 우리 학급 게시판에 업로드 하고, 공유하도록 하였다.



[그림 IV-10] 모듈별 프로그래밍 제작 활동 및 산출물 공유 장면

## 라) 6차시 : 모듈별 홍보자료를 발표하고 평가하기(공유 및 평가)

6차시 수업에서는 이전 차시에 제작하여 엔트리의 학급 게시판에 업로드한 생태도시 홍보 애니메이션을 모듈별로 시연하고 발표하도록 하였다. 발표는 교수자가 화면 공유를 통해 각 모듈의 산출물을 실행하고, 해당 산출물을 제작한 모듈원이 산출물의 주제, 특징, 제작과정, 활동하면서 재밌었던 점, 어려웠던 점 등을 발표하는 형식으로 진행되었다. 교수자와 학급의 다른 모듈원들은 제작물을 함께 관람 후 화상회의 프로그램을 통해 감상과 피드백을 제공하였다. 더불어 다른 모듈의 산출물 하단에 잘된 점, 아쉬운 점, 개선이 필요한 점에 대한 의견을 댓글 형식으로 남기고, 마음에 들거나 추천 하고 싶은 산출물은 ‘마음’을 눌러 점수를 매길 수 있도록 안내하였다.



[그림 IV-11] 모듈별 산출물 발표 장면 및 학습자 간 평가 댓글 화면

수업 후 프로그래밍 소프트웨어를 활용하여 홍보 애니메이션을 제작한 경험을 되돌아보며 성찰일지를 작성하도록 하였다. 성찰일지는 교과목과 소프트웨어를 융합한 수업 및 프로그래밍 소프트웨어(엔트리) 활용 수업에 대해 재미있었던 점, 어려웠던 점, 개선이 필요한 점, 그리고 수업에 대한 전반적 느낌에 대한 질문에 대해 자율적으로 답변할 수 있도록 제작하였으며, 학습자들은 구글 설문지를 통해 답변하였다.

## 나. 교수자 반응 평가

실제 현장 교사가 본 연구에서 도출한 수업설계 모형과 원리를 바탕으로 수업을 설계하고 실행한 후, 이를 토대로 개념 모형, 절차 모형, 설계원리의 사용성을 평가하는 교수자 반응 평가를 실시하였다. 모형의 사용성 평가는 개념 모형과 절차 모형, 설계원리가 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는 데 도움이 되었는지를 묻는 5개의 사용성 평가 문항과 8개의 개방형 문항으로 구성되었다.

교수자 사용성 평가 문항은 개념 모형과 절차 모형, 설계원리와 그에 대한 설명이 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계가 무엇인지 이해하는데 도움이 되었는지, 그리고 수업설계를 위해 무엇을 해야 하는지, 즉 어떤 요소를 고려해야 하는지 이해하는데 도움이 되었는지에 대한 4점 척도 문항으로 이루어져 있다. 현장 평가에 참여한 교수자는 교수자 반응 평가의 모든 항목에서 4점(매우 그렇다)을 부여하였다.

교수자는 개념모형과 그 설명에 대해서는 사회, 국어, 수학 등의 교과와 소프트웨어융합 프로그램이 문제 중심으로 이루어져야 함이 강조되어 있어 처음부터 문제 중심 프로젝트로서 수업을 설계하는 데 도움이 되었다는 의견을 주었다. 또한 학습 지원을 위해 어떤 것을 고려해야 하는지를 꼼꼼히 따져볼 수 있어서 좋았다는 답변을 주었다. 예컨대, 학습 방법 안내, 모듈 구성, 학습자료, 수업 규칙 등 학습 활동을 위해 고려해야 할 구체적인 사항들을 알 수 있도록 짚어주어서 도움이 되었음을 언급하였다. 그리고 학습도구를 선정할 때 어떠한 자원과 도구를 활용해야 하는지를 되짚어 볼 수 있어서 도움이 되었고, 수업 설계 측면과 환경적 측면, 그리고 학습 도구를 종합적으로 고려할 수 있도록 직관적이고 이해하기 쉽도록 개발된 모형이라는 의견을 수렴할 수 있었다. 반면 개념모형과 1:1로 매칭되는 세부 지침이나 예시를 제시해 주면 이해하는 데 보다 도움이 될 것이라는 개선점에 대한 의견도 있었다.

절차모형과 그 설명에 대해서는 개념모형에 있는 다양한 요소를 어떠

한 순서로 고려하여 수업을 설계해야 하는지를 단계별로 자세히 알 수 있다는 점이 좋았다는 의견이 있었다. 그리고 요구분석을 통해 교육과정, 학습 환경 및 학습 자원, 학습자에 대한 분석을 실시한다는 점과, 사용성 평가를 통해 지속적으로 수업을 수정 및 보완한다는 점이 강조되어 있어 도움이 되었고, 이러한 절차모형을 활용하여 보다 완성도 있는 수업 설계가 가능할 것이라는 의견을 주었다. 또한 교육과정 재구성, 프로토타입 개발, 창의적 교과-소프트웨어융합 수업활동 설계가 일련의 선형적인 과정이 아닌, 필요에 따라 순서를 바꾸거나 왔다갔다 할 수 있도록 모형을 구성한 점이 실제 현장에서 수업 설계를 하는 과정을 잘 반영한 것 같으며, 이러한 측면이 본 모형의 강점인 것 같다는 의견도 있었다. 반면 절차 모형의 학습도구 선정 단계 이외에도 소프트웨어 교육의 특징을 드러내는 요소를 투입하여 본 연구의 특징을 좀 더 드러내면 좋을 것 같다는 개선점도 제시되었다.

설계원리와 상세지침에 대해서는 수업에 활용할 수 있는 다양한 사례와 아이디어 발산·수렴을 위한 온·오프라인 도구를 제시하여 수업 설계에 많은 도움이 되었다는 의견을 주었다. 또한 설계원리를 통해 수업 설계 시 구성주의적 관점에서 수업을 설계해야 한다는 점, 창의적 문제해결력을 극대화 하는 목적으로 수업을 설계해야 한다는 점을 새롭게 알 수 있어 시야가 넓어졌다는 의견을 제시하였다. 반면 상세지침 10.1에서 제시된 스토리보드 형태의 활동지를 학생들에게 제공하였을 때 초보 단계 학생들은 미리 블록을 계획해 보며 차근차근 알고리즘을 만들어갈 수 있는 장점이 있는데 반해, 일정 수준 이상의 절차적 사고가 가능한 학생들은 이 작업이 형식적인 절차로 느껴질 수 있기 때문에 학생들의 수준 차이를 고려하여 어떻게 효과적으로 수업을 설계할 수 있는지에 대한 대안을 제시해 주면 좋을 것 같다는 의견을 주었다. 또한 상세지침 7.1에 대해서도 협력 활동을 할 때 프로그래밍을 잘하고 익숙한 학생과 프로그래밍을 처음 접하여 상대적으로 이해도가 낮은 학생을 한 모둠으로 지정하면 프로그래밍 작성 수준이 높은 학생이 전체적인 과정을 주관하고 산출물을 만들 때도 도맡아서 제작할 확률이 높았고, 오히려 비슷한 수준

의 학생이 협력을 할 때 의사소통이 더 잘 이루어졌다는 교수자의 경험을 제시하며 수준차가 나는 학생보다 수준이 비슷한 학생끼리 모둠을 구성하는 것이 더 좋을 것 같다는 의견을 주었다. 더불어 경험이 풍부한 학생과 그렇지 않은 학생을 나눌 수 있는 기준이 명확히 제시되면 수업 설계에 도움이 될 것이라는 추가적인 의견을 제시하였다.

개념모형, 절차모형, 설계원리와 상세지침에 대한 의견과 함께 수업 설계 시 어렵거나 이해하기 어려웠던 부분에 대해서는 발산적 사고도와 수렴적 사고도를 처음 접하는 교수자들을 위해 이를 활용한 구체적인 수업 장면들을 예시에 함께 제시해주면 좋을 것 같다는 의견을 주었다. 예를 들어 PMI를 활용할 때는 댓글을 달 때 활용하거나 피드백을 구두로 전달할 때 활용할 수 있다는 점 등 구체적인 수업 장면을 제시할 필요성이 있다는 의견이었다. 또한 평가도구 측면에서도 구체적으로 어떠한 도구를 활용하면 좋을지, 어떤 평가 요소가 있을지 예시와 사례를 제시하면 수업 설계 시 참고가 될 것이라고 하였다.

## 다. 학습자 반응 평가

### 1) 창의적 문제해결력, 학업적 자기효능감에 대한 학습자 반응

창의적 문제해결을 위한 교과-소프트웨어융합 수업 모형에 대한 효과를 확인하기 위해 창의적 문제해결력, 학업적 자기효능감에 대한 단일집단 사전사후 검사를 실시하였다. 창의적 문제해결력 측정 검사는 총 20 문항으로 한국교육개발원(2001)이 발간한 ‘간편 창의적 문제해결력 검사 개발 연구(2001)’를 기반으로 서울대학교 심리연구실 MI연구팀(2004)에서 개발한 것을 사용하였다. 학업적 자기효능감 확인을 위해서는 김아영, 박인영(2001)의 학업적 자기효능감 척도의 문항을 사용하였다.

수업이 종료된 후 실시한 사후 검사 결과, 창의적 문제해결력과 학업적 자기효능감에 대한 평균 점수가 모두 향상되었음을 확인할 수 있었다.

<표 IV-24> 창의적 문제해결력, 학업적 자기효능감 사전·사후 검사 결과

영역	사례수	사전		사후	
		평균	표준편차	평균	표준편차
창의적 문제해결력	23	3.06	0.68	3.52	0.50
학업적 자기효능감	23	3.22	0.61	3.56	0.47

<표 IV-25> 창의적 문제해결력, 학업적 자기효능감 대응표본 t-검정 결과

영역	사후-사전				
	평균차이	표준편차	t	자유도	유의확률 (양쪽)
창의적 문제해결력	0.45	0.80	2.717	22	.013
학업적 자기효능감	0.35	0.72	1.945	22	.065

p<.05

대응표본 t 검정 결과, 창의적 문제해결력을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업의 실시 전과 실시 후에 측정한 학습자의 창의적 문제해결력은 유의수준 .05에서 유의한 차이를 보였으나, 학업적 자기효능감에서는 유의수준 .05에서 유의한 차이를 보이지 않았다. 따라서 본 모형을 적용한 수업이 학습자의 창의적 문제해결력의 변화에 긍정적인 영향을 미칠 수 있다고 해석할 수 있다.

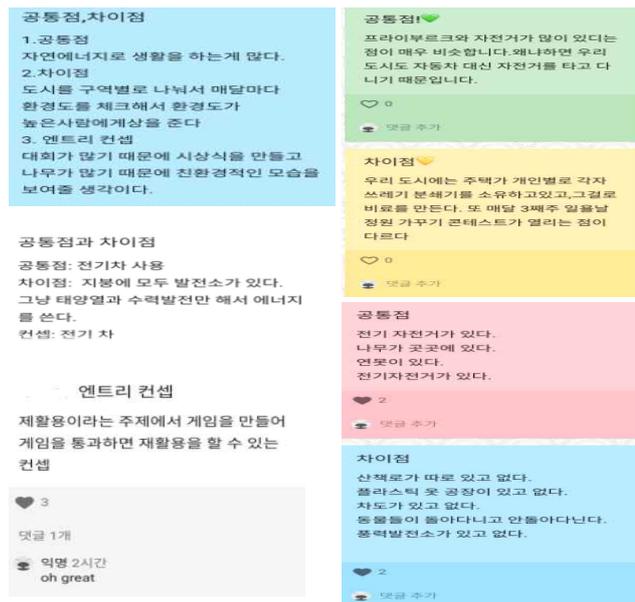
## 2) 수업을 통한 학습자 활동 결과

수업의 각 차시별로 이루어진 창의적 문제해결을 위한 교과-소프트웨어융합 수업 활동의 학습자 산출물을 수집하였다. 수업은 각 차시별로 줌(ZOOM)을 통해 녹화되었으며, 패들릿, 구글독스, 엔트리를 활용하여

제작된 학습자 산출물은 학급 게시판에 링크로 게시되었다. 각 수업에 대한 학습자 활동 결과 및 산출물은 다음과 같다.

### 가) 문제정의 및 아이디어 생성

1차시~2차시 수업에서는 문제인식 및 아이디어 생성을 위한 수업 활동을 전개하였다. 우선 생태도시 홍보자료의 토대를 마련하기 위해서 본 수업 이전에 설계한 학습자의 생태도시를 실제 다른 생태도시와 비교하고 공통점, 차이점을 비교 후, 생태도시의 특징을 분석하여 문제를 정의하는 과정을 수행하였다. 특히 문제인식 및 정의 과정에서는 관련 자료를 수집, 분석하여 패들렛에 추가함으로써 학습자들은 자신들의 사고과정을 정리하여 시각화하였으며, 이를 통해 창의적 문제해결을 위한 문제인식 및 정의 과정을 확인할 수 있었다. 실제 학습자들이 패들렛에 정리한 공통점과 차이점, 엔트리 컨셉에 대한 활동 장면은 아래 [그림 IV-12]와 같다.



[그림 IV-12] 생태도시와의 공통점과 차이점을 바탕으로 엔트리를 활용한 홍보자료 아이디어를 도출하는 실제 장면 예시

또한 정의한 문제를 토대로 생태도시 홍보를 위한 홍보 애니메이션의 컨셉 아이디어를 작성하는 활동을 실시하는 과정을 통해 학습자들의 창의적 문제해결력을 촉진하고 이에 대한 산출물을 확인할 수 있었다. 학습자들은 교수자가 제공한 애니메이션의 장면, 인물, 대사 등에 대한 피드백을 반영하여 홍보자료 제작을 위한 브레인스토밍 과정을 거쳤다. 이를 통해 홍보자료 컨셉에 대한 창의적이고 구체적인 계획을 학습자별로 다양하게 도출하였고, 이를 패들렛 카드에 추가하였다. 예를 들어 한 학습자 모둠은 생태도시의 차별점 대해서 ‘전기를 많이 사용한다’, ‘매연을 내뿜는 자동차는 진입이 불가능 하다.’, ‘집집마다 태양열 발전기가 있다.’ 등의 아이디어를 도출하였고, 교수자는 이에 대해 좀 더 브레인스토밍을 하며 홍보 애니메이션 제작을 위한 구체적인 아이디어를 작성해 볼 것을 피드백하였다. 해당 학습자 모둠은 이에 아이디어를 보다 정교화하여 ‘휘발유 자동차를 직접 운전하는 것과 전기버스를 타는 것을 비교하기’, ‘전기차를 타고 매연 없는 생태도시 소개하기’, ‘전기차로 매연을 없애는 게임 만들기’ 등의 다양한 아이디어를 발전시키고 구상하였다. 또한 다른 학습자들의 카드에 마음을 누르고 댓글을 추가하며 동료 피드백을 수행하였다([그림 IV-13] 참고).

### 계---획

엔트리\_컨셉  
 엔트리봇이 정원 콘테스트가 열리는날 일과,  
 정원을 가꾸고-- 장면 전환--아침밥 먹고--콘테스트 참여--3등을 함--3초 기다리고--상을 들고 집으로 감--장면전환--점심먹고--쓰레기와 음식물을 분쇄--자전거타고 분리수거--자전거 대여소로 다시 자전거 복귀--장면전환--집으로 --잠!

청정한 환경에 피해가 가는 행동은 즉시 마을회관에 말해서 금지할 수 있다. 금지된 행동을 하는 사람은 추방이다!(죽 더러운 도시에서 죽어가라는 뜻...쿠레헬후엘)●

엔나 이고 계획 7이아나구 6아니양? - 지민스

앗 그림네 알려줘서 고마오♥ - 익명

희희 - 지민스

오 친환경에 피해가 가면 바로 벌을 받는다는점에서 완전거같아! - 익명

### 계획 (8)

엔트리가 정원에 물을 주고 말한다  
 "오늘은 정원 콘테스트가 있어! 그래서 정원을 가꾸어야해~!"  
 물을 준 후 장면 전환후 집으로 들어간다.  
 집에서 행동을 바꾸어 밥을 먹는 행동을 한다.  
 밥을 먹고 말한다.  
 "콘테스트에 참가해야해!"  
 장면 전환-----  
 콘테스트에서 사회자가 정원을 둘러본다  
 사회자가 말한다.  
 "일등은 --입니다!" "이등은 에베벨!" 삼등은.."엔투러봇!"  
 엔트리봇이 상을 받는다.  
 장면 전환-----  
 집으로 가서 밥을 먹는다  
 음식물 분쇄기에 넣고 말한다.  
 "음식은 분쇄하면 비료가 되서 친환경적인듯 ㅇㅇ"  
 분쇄기에서 음식물 꺼내 정원으로 간다

[그림 IV-13] 홍보자료 컨셉에 대한 아이디어 도출 및 학습자간 피드백

더불어 학습자들은 1차시 초반에는 아이디어를 도출하고 작성하는 것에 어려움을 느끼고 교수자의 도움을 요청하는 모습을 보였으나, 수업이 진행될수록 패들렛을 활용해 아이디어를 정리하고 생성하는 활동을 원활히 진행하는 모습을 보였으며, 동일한 맥락의 내용이 학습자 면담에서도 언급되었다. 이를 통해 수업이 진행됨에 따라 학습자들의 창의적 아이디어 생성 활동이 빠르게 촉진되었음을 확인할 수 있다.

#### 나) 스토리보드 및 알고리즘 작성을 통한 논리적인 해결안 도출

학습자들은 본격적인 프로그래밍 활동에 앞서 이전 차시에서 도출한 아이디어를 논리적으로 구성하고 최종적인 홍보자료의 틀을 갖추기 활동인 스토리보드 및 알고리즘 작성을 수행하였으며, 이를 통해 본격적인 홍보자료 제작을 위한 논리적 해결안을 마련하는 것을 확인할 수 있었다. 즉, 본 학습활동을 통해 학습자들이 수업 중 패들렛으로 자유롭게 도출한 아이디어를 하나의 맥락으로 구성하여 이야기를 만들고, 이를 절차적으로 장면별 내용을 구성하는 과정을 거치며 엔트리 활용에 앞서 기본적인 틀을 작성함으로써 논리적 해결안을 마련하는 것을 확인할 수 있었다. 예컨대, [그림 IV-14]의 실제 학습자 모듬의 스토리보드 산출물과 같이 엔트리봇이 산책로를 걸어가며 여러 가지 건물에 대한 설명을 하는 장면에 대해서는 ‘집에서 일어난 뒤 산책하러 산책로로 향한다.’, ‘산책하며 건물들을 설명한다.’로 자연어 문장으로 설명 후 등장하는 오브젝트(침실 배경, 산책로 배경, 등장 캐릭터, 주변 사물 오브젝트 등)를 생각해 보고 이를 배치하는 것으로 일련의 절차적인 문제해결 과정을 수업 활동을 통해 경험하는 과정을 확인할 수 있었다.

환경 문제를 알리는 홍보 애니메이션 스토리보드

배경	등장 오브젝트	장면 설명	시간
산책로 기관	집 안 침실, 엔트리 봇	집에서 일어난 뒤 산책하러 산책로로 향한다. 산책하며 기관(?)들을 설명	30초
	산책로, 산책하는 사람들, 쓰레기통		15초
	여러 건물, 기관 등		1분
화단 자전거 대여소	나무, 꽃 등의 식물	주민들이 식물을 심는 모습	30초
	자전거	자전거 빌리는 모습	30초

[그림 IV-14] 스토리보드 산출물

다) 컴퓨팅 사고력 바탕의 프로그래밍 활동을 통한 창의적 문제해결

스토리보드를 토대로 환경 문제를 알리는 홍보 애니메이션을 엔트리를 활용하여 제작하였다. 교수자는 엔트리의 학급 개설 기능을 활용하여 학습자들이 산출물을 업로드 할 수 있는 온라인 학급을 개설하였다. 학습자들은 이전 차시에 구글독스에 작성한 스토리보드를 바탕으로 코딩 소프트웨어를 활용하여 홍보 애니메이션을 제작하였다. 학습 활동은 모듈별로 소회의실을 개설하여 이루어졌으며, 직접 코딩 작업을 수행하는 운전자 역할을 담당하는 학습자가 줌(ZOOM)의 화면 공유를 하고, 나머지 학습자들이 네비게이터 역할을 담당하여 전체적인 스토리를 구성하고 코딩의 논리구조를 확인하였다.

프로그래밍을 통한 창의적 문제해결 과정은 먼저 학습자들이 구상한 아이디어를 구현하기 위해 다양한 자료를 탐색하고 이를 배치하는 과정에서 드러났다. 특히 학습자들은 애니메이션을 효과적으로 표현할 수 있는 다양한 배경 및 사물 오브젝트, 소리 및 음성 자료를 인터넷 검색 후 논의를 통해 선정하는 과정을 거쳤다. 또한 장면 연출을 위한 등장인물

의 대사와 다양한 명령어를 탐색하는 과정에서도 창의적 문제해결 과정을 확인할 수 있었다. 특히 스토리를 이끌어 가기 위한 오브젝트의 움직임 및 대사를 표현하기 위해 코딩 블록 선정, 논리 구조화 작업, 코딩 블록 조립과정을 거치며 프로그래밍을 통한 창의적 문제해결 활동을 수행하였다. 더불어 구성된 아이디어를 다양한 시행착오와 시도를 통해 최적안을 선택하고 이를 해결하는 제작 과정을 거치며 활동 중간중간 작업물을 실행해 보면서 코딩의 오류 및 스토리 진행 상 어색해 보이는 부분을 찾아내고 이를 다시 수정하며 점진적, 반복적인 디버깅 작업을 수행함으로써 컴퓨팅 사고력을 반영한 수업 활동을 제시하였다. 이러한 창의적 교과-소프트웨어 수업 활동을 통해 학습자들은 다양하고 창의적인 산출물을 제작하였다. 예컨대 클릭하면 분리수거가 실행되는 미니 게임이 포함된 애니메이션, 주요 캐릭터가 등장하여 환경도시의 특징과 시설을 안내하는 애니메이션, 환경도시에 살고 있는 학습자의 아바타가 하루를 보내며 환경도시 거주자가 일상에서 접할 수 있는 환경 보호 정책 및 활동을 경험하는 애니메이션 등 다양한 주제의 산출물을 확인할 수 있었다.



[그림 IV-15] 환경보호를 위한 생태도시 홍보 학습자 산출물

## 라) 모듈별 산출물에 대한 다양한 의견 및 피드백 제시

모듈별 산출물을 시연하고 발표하는 활동을 수행하였다. 교수자는 엔트리 학급에 업로드한 모듈별 산출물을 실행하고, 해당 모듈의 학습자들이 산출물에 대해 발표하였다. 학습자들은 줌(ZOOM)을 통해 감상과 피드백을 제공하고, 다른 모듈의 산출물에 댓글을 작성하였다. 댓글은 산출물의 좋은 점, 아쉬운 점, 흥미로운 점을 자유롭게 작성하였으며, 마음에 들거나 추천하고 싶은 산출물에 ‘마음(좋아요)’을 누르도록 안내하였다. 산출물에 대한 학습자 댓글 및 마음(좋아요)이 표시된 화면은 [그림 IV-16]와 같다.

특히 일부 학습자는 교수자와 동료 피드백을 반영하여 자신의 모듈이 제작한 산출물을 개선하는 작업을 수행하였다. 학습자들은 배경 오브젝트를 다양하게 바꾸거나 오브젝트의 움직임과 대사가 나오는 타이밍을 조정하는 등의 수정 작업을 하였다.



[그림 IV-16] 학습자들의 댓글 및 마음(좋아요) 현황

### 3) 학습자 면담 결과

수업에 대한 학습자의 의견을 심층적으로 살펴보기 위해 심층 면담을 수행하였다. 면담은 다섯 명의 학습자를 대상으로 이루어졌으며, 반구조화된 면담의 형식으로 진행되었다. 특히 기존의 소프트웨어 활용 중심 수업에 비해 창의적 문제해결을 위한 교과-소프트웨어 융합 수업이 갖는 의미에 대해 살펴보기 위하여 수업의 주요 활동에 대한 학습자의 의견을 분석하였다. 면담은 소프트웨어 융합 수업에 대한 경험, 소프트웨어 융합 수업의 장점, 어려운 점, 개선이 필요한 점, 수업을 하면서 기억에 남는 활동 및 전반적인 소프트웨어 융합 수업에 대한 태도 및 각 수업 활동에 대한 의견을 묻는 질문으로 구성되었으며, 이에 대한 학습자의 의견은 다음과 같다.

우선 과거 다른 과목과 소프트웨어를 융합한 수업의 경험 여부 및 해당 수업과 지금 수업의 차이점을 면담을 통해 확인하였다. 학생들은 소프트웨어를 활용하는 동아리 활동을 한 경험은 있지만 당시 활동에서는 EPL 소프트웨어를 중심으로 활용법을 배우거나 소프트웨어 활용 예제를 만들고 시연해 보는 수준의 활동이었다고 답했다. 그러나 이번 수업은 사회과목의 문제를 토대로 이에 대한 자료를 수집, 분석하고, 아이디어를 생성한 후 스토리보드를 활용하여 엔트리로 구현하는 활동을 함으로써 본격적으로 소프트웨어를 다루면서 종합적인 융합 교과 내용을 학습할 수 있었다는 의견을 확인할 수 있었다. 이와 관련한 학습자 의견은 다음과 같다.

학습자 A : 이런 수업을 해본적은 없다. 엔트리를 배우기는 했는데 엔트리를 어떻게 쓰는지 내용만 배우고 이번 수업처럼 여러 활동을 하는 수업은 없었다.

학습자 B : 처음부터 아이디어를 짜내고 스토리보드에 적어보고 엔트리로 움직여보는 과정이 나름대로 환경학습에 대해 한 번 더 생각해 볼 수 있는 경험이 되었다.

학습자 C : 아이디어를 뽑고 스토리보드를 써보는 것을 이전에 조금 해봤지만, 이번에 제대로 해봤다. 특히 동아리 활동 말고 학교 수업에서 엔트리를 쓰면서 수업 하는 건 처음이라 재밌었다.

학습자 E : 엔트리는 사용 안 해봤고, 동아리 활동 때 글쓰기를 구글독스로 해봤다. 근데 이번처럼 프로그래밍 소프트웨어를 본격적으로 써본 건 처음이다.

수업이 진행되는 동안 학습자들은 자택에서 개인 컴퓨터를 통해 온라인으로 학습하였으며, 학습활동의 많은 부분이 패들렛이나 구글독스 등 창의적 문제해결을 위한 수렴·발산적 사고 도구와 엔트리 등의 프로그래밍 소프트웨어를 활용하여 이루어졌다. 면담에 참여한 학생들은 수업에서 이와 같은 전자기기 및 소프트웨어를 활용한 것이 학습에 대한 흥미를 높이고 동기를 부여하는 데 영향을 주었다고 하였다.

학습자 A : 패들렛에 아이디어를 쓰고, 아이디어를 가지고 구글독스에 스토리보드를 만들었는데 처음엔 조금 어렵다가 하다 보니 생각도 잘 나고 그걸 또 스토리보드로 정리할 수 있어서 재밌었다.

학습자 C : 계속 컴퓨터를 활용해서 수업하는 것이 재밌었다. 패들렛을 활용하니 아이디어를 좀 더 쉽게 짜낼 수 있었다. 프로그래밍도 할 수 있고 엔트리에서 직접 코딩을 하니까 캐릭터가 움직여서 신기하고 재밌었다.

한편으로는 컴퓨터를 다루는 것에 익숙하지 않거나 엔트리를 처음 해보는 학습자들이 있어 이에 대한 염려를 표현한 면담자들의 의견도 확인할 수 있었다. 이와 같은 학습자 간의 수준 차이를 고려하여 향후에는 선수학습을 강화하거나 동아리 활동 및 정규 교과 차원에서 소프트웨어 융합 수업을 실시하는 방안에 대해 의견을 준 학습자도 있었다.

학습자 C : 전자기기를 잘 못 다루는 친구들은 이런 수업을 별로 좋아하지 않는다. 호불호가 갈린다.

학습자 D : 제 생각엔 일단 이게 중요한 학습이고, 4차 산업혁명일 중요해서 배우는 건 맞는 것 같은데 일단 엔트리를 처음 접해보는 학생은 싫어하는 학생들도 꽤 있었다.

학습자 E : 저는 엔트리가 처음이라 좀 어려웠다. 어렸을 때, 저학년 때부터 동아리 활동이나 학교 수업에서 조금씩 나눠서 배울 수 있었으면 좋겠다. 이번에는 좀 몰아서 배운 것 같다.

또한 수업 활동 중 아이디어를 생성하고 브레인스토밍을 하는 과정에 대해서 학습자들의 긍정적인 반응을 확인하였다. 학습자들은 수업 초반 아이디어를 내는 데에 어려움을 겪었지만, 지속적인 브레인스토밍과 이를 지원하는 소프트웨어 도구를 활용하여 후반부에는 다양한 아이디어를 도출할 수 있었다는 의견을 주었다. 또한 아이디어를 내기 위해 환경문제와 환경 도시에 대한 심층적인 자료 조사를 할 수 있어서 학습에 도움이 되었다는 의견을 확인하였다.

학습자 A : 맨 처음에는 아이디어를 짜내는 것이 어려웠는데, 패들렛에 적어보고 자료도 찾아보고 생각을 계속 하니 아이디어를 낼 수 있었다. 계속 생각하다보니 머리가 더 잘 돌아갔다.

학습자 B : 패들렛에 하나하나 아이디어를 적어서 좀 더 생각이 잘났다. 생각이 하나로 뭉쳐있었는데 그걸 풀어서 조금씩 써서 그런 것 같다.

학습자 C : 아이디어 만들 때 패들렛을 쓰면서 좀 더 세세하게 생각해 볼 수 있는 시간이 많아져서 더 좋았던 것 같다.

학습자 D : 아이디어를 내려고 환경문제를 조사해봤는데 일단 저희가 꼭 알아야할 환경문제에 대한 배경지식을 알 수 있어

서 공부하는데 도움이 되었다.

스토리보드를 작성하고 엔트리로 직접 구현해보는 학습 활동에 대해서는 다양하게 생성한 아이디어를 엮어 하나의 이야기로 만들어가는 과정이 재미있었다는 의견을 확인하였다. 특히 학습자들 스스로 주제와 상황, 등장인물과 대사를 정해 이야기의 토대를 만들고 스토리보드를 제작해 가는 과정이 재미있고 유익했다는 긍정적인 반응이 많았다. 또한 처음부터 스토리보드라는 큰 틀을 만들고 이를 엔트리로 구현할 수 있어 효율적이었다는 의견을 확인할 수 있었다.

학습자 A : 패들렛에 적은 아이디어들을 엮어서 스토리로 만들어 내는 게 도움이 되었다. 스토리 보드를 만들 때도 예전에 해본 연극활동에서 처럼 계속 장면이나 대사, 등장인물들을 정하고 생각을 정리해서 대본처럼 적을 수 있어서 익숙하고 재밌었다.

학습자 C : 맨 처음부터 그냥 하다보면 내가 원래 생각했던 것에서 다르게 될 수도 있는데 아예 처음부터 틀을 잡아놓으니 엔트리를 더 잘 만들 수 있었던 것 같다.

학습자 D : 스토리보드가 있어서 더 좋았다. 배경도 장면도 다 적어놓고 하니까 옮기기가 더 좋았다.

학습자 E : 스토리보드가 있었던 게 훨씬 더 좋았다. 만들다보니까 생각했던 것 보다 훨씬 쉽게 할 수 있어서 힘들지도 않고 재밌었다

한편 학습자들은 과제 수행을 위해 모둠을 나누고 서로 협력했던 경험에 대해 긍정적인 반응을 보였다. 우선 아이디어 생성 시 혼자보다 함께 생각할 수 있어서 더 신선하고 다양한 아이디어를 생각해 낼 수 있었고, 모둠원별로 역할을 정해 역할 별로 담당을 나누어 엔트리를 만들어 볼 수 있어서 효율적으로 과제를 마칠 수 있었다는 답변을 확인할 수 있

었다. 더불어 프로그래밍 소프트웨어를 잘 다루지 못해 어려움을 겪는 학생도 모둠원과의 협력을 통해 프로젝트에 참여할 수 있었다는 점도 긍정적인 부분으로 꼽혔다.

학습자 A : 아이디어가 한명이서 내는 것 보다 두 명이서 내는 것이 더 빨리 많이 생각나서 좋았고, 또 역할이 두 개로 나뉘지다 보니까 한 사람이 들어야 할 짐이 더 적어져서 편했던 것 같다.

학습자 B : 일단 혼자 생각하는 것 보다 두 명이 있어서 더 생각하기 편했다.

학습자 D : 같이 팀으로 학습하는 게 괜찮은 거 같은 게 일단 저희 반에 엔트리를 처음해서 어려움을 겪는 학생도 많은데 팀으로 하니까 그래도 그 프로젝트를 같이 할 수 있었고, 더 빨리 같이 완성할 수 있었다.

그러나 과제 진행 시 모둠원끼리의 의견이 맞지 않아 힘들었거나, 프로그래밍 수준에 따라 역할이 고정될 수밖에 없는 등의 아쉬운 점에 대한 의견도 확인할 수 있었다. 더불어 온라인 환경에서의 모둠활동의 어려움을 지적하는 면담 의견도 확인할 수 있었다.

학습자 B : 두 명 이서 하니까 혼자 하는 것 보다 의견 안 맞을 때가 많아서 좀 오래 걸린 것 같다.

학습자 A : 계속 하다보니까 아이디어를 내는 역할과 엔트리를 만드는 역할을 바꾸지는 못했는데, 이건 엔트리를 잘 못하는 애들이 있어서 바꾸면 더 어려울 것 같다. 좀 더 익숙해 진다면 서로 코딩한 걸 활용해서 만들어 볼 수 있을 것 같다.

학습자 D : 팀으로 학습 했을 때는 한 사람이 만들게 된다. 저희가 아이디어를 주는 건 좋은 것 같은데, 일단 여기서는 소프트웨어 학습에 집중해야 되는데 그 부분에서는 한 학생밖에

프로그램에 접할 수 없어서 살짝 아쉬움이 있었다.

학습자 E : 온라인(ZOOM)으로 한 사람만 화면 공유를 하게 되니까 계속 그 사람만 코딩을 하게 된다. 엔트리를 같이 하기에 그렇게 좋은 환경이 아닌 것 같았다.

산출물을 공유하고 학습자들 간 평가하는 활동에 대해서는 다른 학습자의 의견을 참고하여 자신의 산출물을 더 개선할 수 있고, 자신도 다른 학습자들에게 피드백을 줄 수 있어서 유익했다는 의견을 확인하였다. 그러나 처음 해보는 학생들의 산출물에 하트 수가 적어 학생들의 선수학습 차이에 의한 형평성을 고려할 필요가 있다는 의견도 확인할 수 있었다.

학습자 C : 서로 의견을 자꾸 쓰다보니까 더 수정할 수 있고 제 작품에 대한 다른 사람의 의견을 들어보니까 더 자신을 돌아보게 되는 게 있었다. 다음에 만들 때는 이렇게 만들어야 할 거같다라고 생각할 수 있었다.

학습자 E : 고칠 점이랑 개선점을 말해주는 것이 좋았다.

학습자 C : 마지막에 엔트리 작품 만들고 서로 공유할 때가 재밌었다. 다른 애들 작품을 보고 평가하는 것도 재밌었고 다른 애들이 제 작품을 평가해주는 것도 좋았다.

학습자 D : 댓글에 잘한 점, 아쉬운 점을 쓰는 것은 그걸 발전시킬 수 있으니까 좋았다고 생각하는데, 하트 누르는 것에 대해서는 저희가 처음 해보는 학생들이 꽤 있다. 그 학생들은 조작하는 거에 대해 미숙하니까 잘 못했을 수도 있는데 일단 하트를 받는 사람만 많이 받고 못 받은 사람만 많이 못 받아서 어떤 친구들에겐 좀 아쉬워서 이것 어떻게 해야 하나 걱정했다.

또한 다양한 교과목과 소프트웨어를 융합한 교과-소프트웨어융합 수업에 대해 학생들의 긍정적인 반응을 확인할 수 있었다. 특히 수업의 핵

심 내용이었던 환경 문제와 생태 도시 조성에 대해 알아보고, 이를 소프트웨어와 접목하여 배울 수 있었던 점이 학습에 대한 흥미를 증가시켰다는 학습자들의 의견을 확인할 수 있었다.

학습자 C : 여러 가지 과목을 같이 배우니까 다양한 각도에서 문제를 보고 해결할 수 있는 것 같다

학습자 D : 제 생각에서는 환경문제에 대해 생각해 볼 수 있어서 좋았다고 생각하는 게 이런 거에 대해 안배우면 많이 접해볼 수 없다. 근데 이런 문제를 생각해서 자료도 모으고 아이디어를 짜고 그걸 스토리보드로 작성해서 엔트리로 애니메이션을 만들면서 일단 조사를 많이 해야 하고 일단 그거에 대해 알아갈 수 있어서 좋았던 것 같다

학습자 C : 환경을 맨날 지켜야 한다는 것은 매년 했던 건데 엔트리로 한다는 게 새롭고 재밌었고, 앞으로도 비슷한 문제에 대해 더 나아가서 스크래치나 파이썬을 활용해 다른 과목이랑 융합적으로 수업해보고 싶다.

학습자 E : 코딩 실력이 늘어서 좋았고 같이 생태도시를 공부할 수 있어서 좋았다. 꿈이 하나 더 늘었다. 도시계획사다. 환경도시를 꾸미고 싶다.

또한 앞으로 과학과 같은 다른 교과와 소프트웨어 융합 과목도 배우고 싶다는 의견도 많아 학습자들의 교과-소프트웨어융합 수업에 대한 학습 동기가 상승한 것을 확인하였다.

학습자 C : 저는 과학이랑 융합하면 재밌을 것 같다. 실험을 엔트리로 구현해도 재밌을 것 같고 선생님 수업할 때 PPT를 활용하시는데 엔트리를 활용해도 좋을 것 같다.

학습자 D : 다음에는 과학과 융합한 소프트웨어 수업을 들어보고 싶다. 일단 지금 저희가 뼈가 어떻게 움직이는지를 배우

고 있는데 코딩으로 하면 실제처럼 그 동작을 볼 수 있으니까, 그냥 PPT로 하면 이해하기 어려울 수 있는데 엔트리를 활용하면 그걸 더 이해하기 쉽게 바꿀 수 있고 과학실에서 실험하듯이 보는 게 원격에서는 최대한 살릴 수 있을 거 같다

가장 기억에 남는 활동에 대한 질문에 대해서 학습자들은 프로그래밍 소프트웨어를 활용하여 직접 산출물을 만들어볼 수 있었던 활동과 함께 스토리를 만들었던 활동을 꼽았다. 특히 프로그래밍 활동의 경우 코딩으로 만든 과제물을 반복적으로 플레이해 보고, 고칠 점을 찾아 수정해 가는 과정에서 성취감을 느꼈다는 학습자의 의견을 확인할 수 있었다.

학습자 A : 엔트리로 설명하는 것을 만든 게 제일 기억이 많이 남는다. 직접 코딩을 하면서 만들어봐서 더 재밌고 즐거웠다.

학습자 D : 스토리보드를 만들고 그걸 엔트리를 만드는 과정이 제일 재밌었다. 왜냐하면 일단 코딩으로 계속 만들고 플레이해보고 하면서 안되는 점을 고쳤는데, 계속 고치고 완성품을 만들어가면서 성취감을 느꼈다. 그걸 스토리보드부터 짜고 만드는 점이 제일 재밌었다.

학습자 E : 저는 스토리 만들어서 애들이랑 얘기하는 게 제일 좋았다. 얘기하면서 또 새로운 스토리가 나와서 계속 재밌게 만들 수 있었다.

#### 4. 최종 모형

본 연구는 컴퓨팅 사고를 바탕으로 기존 교과목을 소프트웨어와 접목시킨 교과-소프트웨어융합 수업을 효과적으로 설계하기 위한 수업설계 모형과 원리를 개발하고자 하는 것을 목적으로 한다.

본 모형은 기존 교과와 소프트웨어를 융합하여 학습자들의 창의적 문제해결력을 개발하기 위한 목적으로 실시하는 수업에 활용할 수 있다. 본 모형은 주 사용자를 기초적인 수준의 소프트웨어와 코딩 지식을 가진 초등학교 교사로 가정하고 있으며, 원활한 수업 진행을 위해 하드웨어 및 소프트웨어, 인터넷 연결 인프라를 갖춘 학교에서 진행되는 것을 전제로 한다. 또한 본 모형은 기존 교과와 소프트웨어의 다학문적 융합을 바탕으로 소프트웨어가 학습활동의 주요 학습도구로 사용되는 것을 가정한다. 본 모형을 통해 초등학교 맥락의 교과-소프트웨어융합 수업을 위한 융합 문제의 개발, 수업 활동의 구성과 이를 지원하는 소프트웨어 도구 및 사고지원도구의 활용 방법을 전체적인 수업설계 과정 속에서 체계적으로 파악할 수 있다. 이상의 모형의 가정을 정리하면 <표 IV-26>와 같다.

<표 IV-26> 모형의 가정

구분	모형의 가정
주 사용자	<ul style="list-style-type: none"> <li>기초적인 수준의 소프트웨어와 코딩지식을 가진 초등학교 교사</li> </ul>
사용 환경	<ul style="list-style-type: none"> <li>하드웨어(PC, 태블릿PC 등) 및 소프트웨어(코딩 프로그램, 학습 플랫폼 등), 인터넷 연결 장비 및 인프라를 갖춘 학습 환경</li> </ul>
융합 형태	<ul style="list-style-type: none"> <li>기존 교과 지식과 소프트웨어의 다학문적 융합 형태</li> <li>소프트웨어가 학습활동의 주요 학습도구로 사용되는 것을 가정</li> </ul>

본 모형의 개발을 위하여 선행문헌 고찰을 통해 도출된 초기 구성요소, 수업설계 원리 및 상세지침, 개념모형과 절차모형을 두 차례의 전문가 타당화 검토를 통해 수정과 보완을 반복하였다. 최종 모형은 내적 타당화의 사용성 평가 과정에서 확인한 현장 교사의 의견과, 외적 타당화 과정 중 수업 현장 및 학습자와 교수자 평가에서 발견한 사항을 바탕으로 수정사항을 반영하여 도출되었다. 또한 내적 타당화와 외적 타당화 과정 중 전문가 및 교수자의 의견과 함께 관련 선행 연구를 참고하여 연구자의 학문적 고찰을 통해 최종적인 수정 과정을 거쳤다. 최종 수업설계 원리 및 상세지침, 개념 모형, 절차모형에 반영한 수정사항은 다음 <표 IV-27>과 같다.

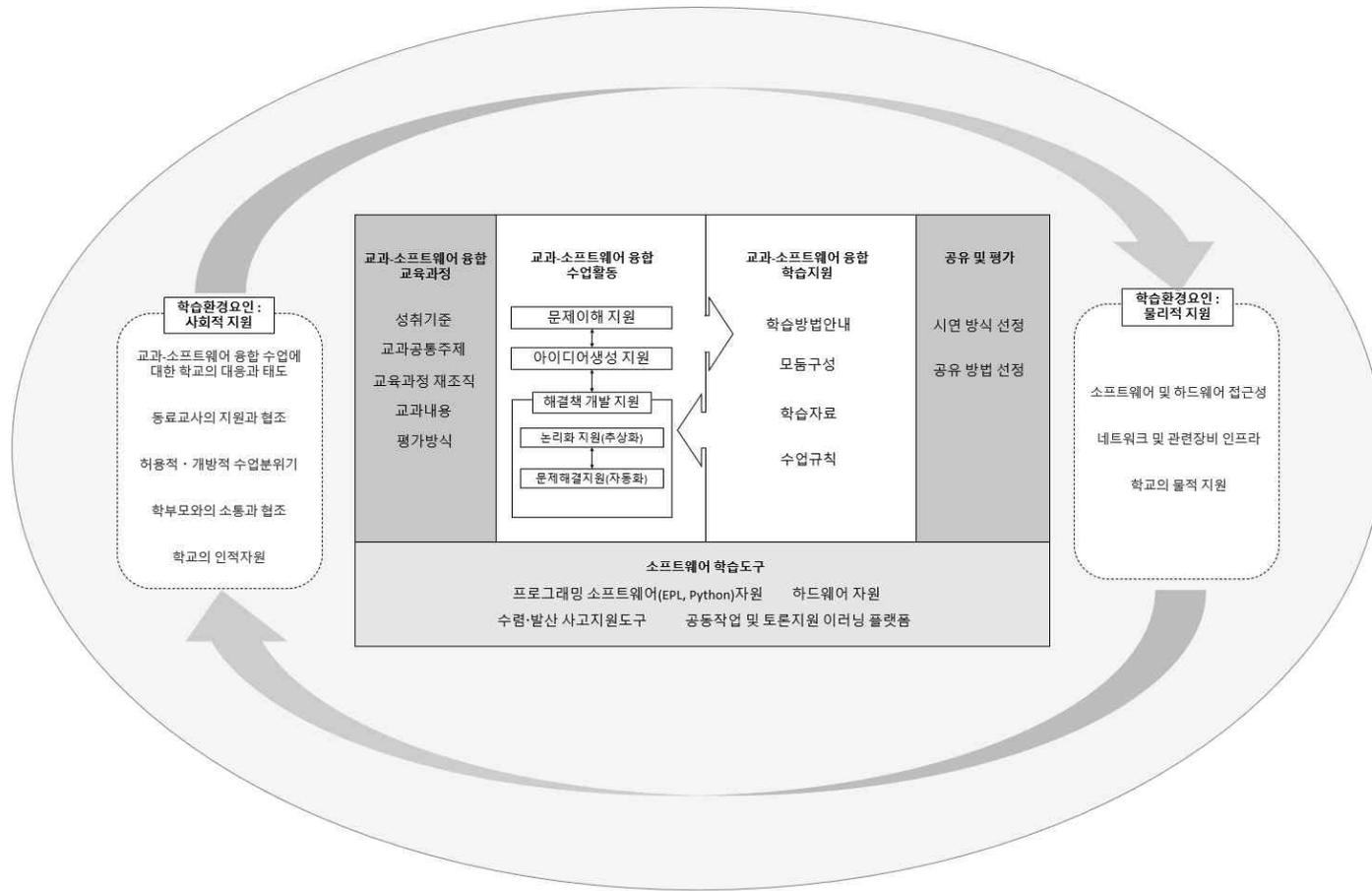
<표 IV-27> 최종 모형에 반영된 수정 사항

	제안 사항	반영 사항
개념모형	학습환경 요인이 다소 복잡해 보이므로 유목화를 통해 정리할 것	학습환경 요인을 사회적 요인과 물리적 요인으로 구분하고 유목화하여 제시함
	구성요소인 ‘공유 및 평가’는 이미 평가기준이 선정된 상태에서 실제 평가가 이루어지는 단계이므로, ‘평가 기준 선정’을 다른 구성요소에 포함시킬 것	‘공유 및 평가’의 하위 요소인 ‘평가 기준 선정’을 ‘교과-소프트웨어융합 교육과정’ 단계로 이동하고 ‘평가 방식’으로 용어 변경
	일부 용어를 초등교사가 이해하기 쉬운 용어로 수정시킬 것	‘프로그래밍 소프트웨어’에 ‘EPL’을 병행 표기함
절차모형	‘프로토타입 개발’ 단계와 ‘프로그램 개발’ 단계의 중첩성을 수정하여 제시할 것	‘프로그램 개발’ 단계의 ‘문제 수정’을 삭제하고 상위 단계로 위계를 수정한 후 단계명을 ‘최종 프로그램 개발’ 단계로 수정

	‘창의적 교과-SW융합수업활동설계’의 상위 위계와 하위 위계명이 중첩되므로 수정하여 제시할 것	상위 위계명을 ‘창의적 교과-소프트웨어융합 수업프로그램 개발’로 수정
설계원리 및 상세지침	현장 교사가 수업 설계 시 참고하면 도움이 될 만한 예시를 제시하고, 익숙한 용어를 활용할 것	일부 상세지침의 예시를 추가하고, 용어를 변경함
	사전에 학습자들의 소프트웨어에 대한 이해와 활용 수준을 파악하고 기초적인 사전 학습이 필요함을 제시할 것	상세지침 1.2를 수정함
	프로그래밍 활동 시 학습자가 지속적, 반복적인 도전을 할 수 있도록 교사가 수행 할 수 있는 방안을 제안할 것	11.1에 관련 내용의 해설을 추가함

## 가. 개념 모형 및 구성요소, 설계원리

창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하기 위해서는 우선 수업을 설계하기 위한 주요 구성요소들과 구성요소들의 관계를 이해할 필요가 있다. 이를 개념화 하여 시각적인 개념 모형으로 나타내면 다음 [그림 IV-17]와 같다.



[그림 IV-17] 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계의 개념 모형(최종)

최종 개념모형은 1) 교과-소프트웨어융합 교육과정, 2) 교과-소프트웨어융합 수업 활동, 3) 교과-소프트웨어융합 학습 지원, 4) 교과-소프트웨어융합 학습 도구, 5) 공유 및 평가의 여섯 가지 요소로 구성되며, 각 구성요소 간의 상호작용을 포함한다.

창의적 문제해결력을 위한 교과-소프트웨어융합 수업 설계는 우선 ‘교과-소프트웨어융합 수업활동’이 중심이 되어 ‘교과-소프트웨어융합 학습 지원’과 상호 지원을 주고받는다. 이러한 수업 활동은 ‘교과-소프트웨어융합 교육과정’에 따라 이루어지며, ‘공유 및 평가’로 마무리된다. 전체적인 수업 활동은 상호 유기적으로 영향을 주고받으며 ‘교과-소프트웨어융합 학습 도구’의 지원을 필요로 한다. 이러한 수업 활동과 도구 지원은 사회적, 물리적인 학습 환경 요인을 토대로 진행될 수 있다.

각 구성요소별 세부사항을 살펴보면 다음과 같다.

우선 ‘교과-소프트웨어융합 교육과정’은 교수자가 소프트웨어와 융합할 과목을 선정하고 수업의 주제 및 목표를 정해야 함을 알려준다. 이때 교수는 하나의 맥락에서 전체적인 수업이 이루어질 수 있도록 융합할 교과목의 단원과 내용, 성취기준을 분석하여 공통적인 주제를 도출한다. 이러한 과정에서 교수는 교과-소프트웨어융합 수업이 성취기준을 달성하면서 목표한 바를 효과적으로 이룰 수 있는지를 파악해야 한다. 교과-소프트웨어융합 수업은 일반적으로 실생활의 문제를 컴퓨팅 사고력을 활용하여 문제를 분석하고 창의적으로 해결할 수 있는 방법을 모색하여 이를 소프트웨어로 구현해 보는 과정으로 진행된다. 학습자들은 복잡한 문제를 해결하기 위해 자유롭게 아이디어를 구상하고 이를 추상화하여 구현하는 과정을 통해 문제를 다양한 측면에서 인식하면서 논리적인 관점으로 해결하는 과정을 경험하게 된다.

중심에 위치하는 ‘교과-소프트웨어융합 수업활동’과 ‘교과-소프트웨어융합 학습지원’은 ‘교과-소프트웨어융합 교육과정’과 상호 영향을 주고받는다. ‘교과-소프트웨어융합 수업활동’은 학습자가 창의적으로 사고하고 문제를 해결하기 위한 활동을 설계하는 것으로, 수업 설계 단계에서 고려해야 할 핵심적인 활동이다. 수업 활동의 하위 요소는 문제이해 지원,

아이디어 생성 지원, 해결책 개발 지원으로 이루어져 있으며, 해결책 개발 지원은 다시 논리화(추상화) 지원과 문제해결(자동화) 지원의 하위요소로 구성되어 있다. 예를 들어 음악 교과와 소프트웨어를 융합하여 우리나라 전통 악보를 만들고 연주해 보는 수업을 설계하는 경우, 학생들은 수업에서 전통 악기와 악보의 중요성을 이해하고(문제 이해), 만들고 싶은 악곡의 아이디어를 자유롭게 구상한 후(아이디어 생성), 자신만의 악보를 만들어(논리화) 코딩 소프트웨어로 구현(자동화)할 수 있다. 이러한 과정은 일방향이 아니며, 수업 활동은 유기적인 순서로 진행된다. 예컨대 학습자는 프로그래밍 소프트웨어로 작성한 악보의 연주를 듣고 다시 악보를 수정하거나 새로운 아이디어를 반영할 수 있다.

‘교과-소프트웨어융합 학습지원’은 수업 진행을 위해 필요한 전반적인 지원사항을 의미한다. 교수자는 수업에 필요한 정보 안내, 학습자료 작성 및 배포, 모둠 활동, 수업에 대한 규칙을 안내하는 등의 수업 지원 사항을 확인하고 원활한 수업 진행을 할 수 있도록 고려해야 한다. 정보 안내의 경우, 안내의 내용과 형식, 게시 형태를 고려하여 학습자가 이해하기 쉽고 언제든 열람할 수 있도록 안내문을 작성하고 공지할 필요가 있다. 모둠활동에 대한 경우도 수업 활동에 알맞은 모둠원 수와 학습자 수준 차이를 고려한 모둠 구성 등에 대한 사항을 고려하여 수업을 설계해야 한다. 이러한 학습 지원은 ‘교과-소프트웨어융합 수업활동’과 상호작용하며 다양한 측면에서 수업 활동에 적합한 지원을 제공한다.

‘교과-소프트웨어융합 학습도구’는 다양한 학습 도구를 계획하고 확보하여 수업이 원활히 진행될 수 있도록 각 수업 활동을 지원하는 것이다. 즉 EPL, 파이썬 등 프로그래밍 소프트웨어와 노트북, 태블릿 PC 등의 하드웨어, 아이디어의 수렴 발산 사고를 지원할 수 있는 패들렛, 코글 등의 사고지원 도구, 그리고 협업 프로젝트 및 토론 활동을 지원하는 이터닝 플랫폼 등을 수업에 활용할 수 있도록 확보하고 이를 바탕으로 도구 활용을 계획하고 수업을 설계하는 것이다.

‘공유 및 평가’는 학습 활동의 산출물의 공유 방식과 평가 방법을 설계하는 것이다. 교수자는 학습자들이 학습활동의 산출물을 어떻게 시연

및 공유하고 어떤 방식으로 평가 및 피드백을 받을 것인지 대해서 계획 하도록 한다. 예를 들어 학습자들은 산출물을 온라인 플랫폼에 게시하거나, PPT를 활용한 발표 자료를 제작하여 학습 활동을 정리하고 이를 온라인 게시판이나 클라우드에 공유하고 교수자 및 다른 학습자들에게 댓글이나 추천의 형태로 피드백을 받을 수 있다.

개념모형의 바깥 부분에 위치한 환경 요인은 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업 설계에 영향을 미치는 바탕 혹은 배경의 의미를 갖는다. 환경 요인은 사회적 지원 요인과 물리적 지원 요인으로 구분할 수 있다. 이 중 사회적 지원 요인은 교과-소프트웨어융합 수업에 대한 학교의 대응과 태도나 동료 교사의 지원과 협조, 허용적 개방적 수업 분위기, 학부모와의 소통과 협조, 학교의 인적 자원 등 교과-소프트웨어융합 수업에 대한 학교 내외의 사회적 협조를 의미한다. 물리적 지원 요인은 소프트웨어 및 하드웨어에 대한 접근성, 유선랜이나 와이파이 등 네트워크 및 관련 장비에 대한 인프라, 그리고 노트북이나 태블릿 PC 등 학교에서 제공하는 물적 지원 등 교과-소프트웨어융합 수업 진행을 위해 구축되어야 할 환경 및 자원을 의미한다. 교수자는 이러한 사회적, 물리적 환경을 고려하여 수업의 진행 가능 여부를 판단할 수 있으며, 각 수업활동은 다양한 환경요인을 반영하여 설계해야 한다.

개념 모형을 통해 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 고려해야 하는 요소들을 확인하였다면, 다음 <표 IV-28>에 제시된 설계원리와 상세지침을 적용할 수 있다.

<표 IV-28> 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계 원리 및 상세지침(최종)

구성요소	원리 및 상세지침	
<b>교과-소프트웨어 융합 교육과정</b>	<b>원리 1. 수업 실행 가능성의 원리</b> 소프트웨어와 연계할 학습내용과 수업 환경을 분석하고 교과-소프트웨어융합 수업의 실행 가능성과 효과성을 판단한다(off).	
	지침 1.1 융합하려는 교과 및 단원의 성취기준, 학습내용, 교수학습 방법, 평가방법을 확인하고 교과-소프트웨어융합수업이 구현 가능한지 판단하라(off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 융합할 교과목과 소프트웨어융합수업을 적용시킬 단원은 무엇인가?</li> <li>• 단원의 성취기준과 학습내용은 무엇인가?</li> <li>• 수업활동과 산출물에 대한 평가도구 및 평가방법은 무엇인가?</li> </ul>
	지침 1.2 학습자의 교과-소프트웨어융합 수업에 대한 이해 수준과 소프트웨어 프로그램 활용 수준을 파악하고 사전에 기초적인 소프트웨어활용 방법을 안내하라(off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 학습자는 교과-소프트웨어융합 수업 내용과 과제를 이해하고 산출물을 만들어낼 수 있는가?</li> <li>• 학습자의 컴퓨터 프로그래밍 소프트웨어(EPL) 조작 수준은 어느 정도인가?</li> <li>• 학습자는 교과-소프트웨어융합 수업에 대한 선행 경험이 있는가?</li> </ul>
	지침 1.3 수업의 실행 및 지원 가능성을 확인하기 위해 학습 환경, 물리적 자원(HW), 학교 방침 및 분위기, 전문가 그룹 등 제반 자원과 환경을 파악하라(off).	

	예시 및 해설	<ul style="list-style-type: none"> <li>• 학교 방침과 문화가 교과-소프트웨어융합수업을 실행하기에 적절한가?</li> <li>• 교과-소프트웨어융합수업 실행을 위해 동료 교사의 이해와 협조가 적절히 이루어지는가?</li> <li>• 교과-소프트웨어융합수업 실행을 위해 자원(시간, 비용)이 얼마나 소요되고, 이를 충당할 수 있는가?</li> <li>• 교과-소프트웨어융합수업을 설계하고 실행하기 위해 조언을 받을 수 있는 전문가 그룹(교과-소프트웨어융합 수업 경험이 있는 동료 교사, 프로그래밍 전문가)이 있는가?</li> </ul>
<p>지침 1.4 프로그래밍 소프트웨어를 활용한 교과융합 수업이 성취 기준 달성에 도움이 되는지를 확인하라(off).</p>		
	예시 및 해설	<ul style="list-style-type: none"> <li>• 교과-소프트웨어융합 수업이 기존 교과 수업보다 상대적으로 어떤 이점이 있는가?</li> <li>• 프로그래밍 소프트웨어를 활용하는 것이 활용하지 않는 수업 보다 어떤 점에서 효과적인가?</li> <li>• 교과-소프트웨어융합 수업을 통해 수업의 성취기준을 달성하면서, 기존 교육과정보다 창의적 문제해결을 위한 지식과 경험을 제공하고 지도하는데 적절한가?</li> </ul>
<p><b>원리 2. 맥락 재구성의 원리</b>  융합할 교과/단원의 학습내용을 바탕으로 교과-소프트웨어융합 교육과정을 하나의 맥락으로 재구성한다(off).</p>		
<p>지침 2.1 교과 특성, 주제의 연계성, 학습자 수준을 고려하여 주제와 관련된 주요 내용을 중심으로 필요시 내용을 생략하거나 단원의 순서를 바꾸도록 하라(off).</p>		
	예시 및 해설	<ul style="list-style-type: none"> <li>• 각 교과의 성취기준을 확인하고 공통 주제를 선정 한 후, 통합된 학습 내용을 중심으로 교육과정 상의 수업 순서를 바꾸거나 주요 내용을 다루는 차시를 늘이고, 주변 차시를 줄여 시수를 확보하고 수업을 재구성한다.</li> </ul>

지침 2.2 융합 주제를 중심으로 융합 할 교과와 소프트웨어 수업의 내용을 하나의 맥락으로 통합하라(off).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 예시 1) 초등학교 5~6학년 음악 : 프로그래밍 소프트웨어를 활용하여 악보를 만들고 연주 해보기</li> <li>• 예시 2) 초등학교 6학년 실과 : ‘자원 관리와 자립’ 단원의 내용을 중심으로 쓰레기 분리 수거를 홍보하는 애니메이션 만들기</li> </ul>
<b>원리 3. 실제적 문제의 원리</b>	
융합 과목의 소프트웨어 연계요소를 고려하여 학습자에게 친근감 있고 도전감 있는 실제적인 교과융합 문제를 설계한다(off).	
지침 3.1 융합주제, 수업목표, 과제 형식을 고려하여 컴퓨팅 사고를 활용하여 해결할 수 있는 실제 생활의 문제를 선정하라(off).	
예시 및 해설	<p>○ [6사05-03] 초등학교 6학년 사회과목의 ‘우리나라의 정치 발전’ 단원 중 ‘일상생활과 민주주의’</p> <ul style="list-style-type: none"> <li>• 성취기준 : 일상생활에서 경험하는 민주주의 실천 사례를 탐구하여 민주주의의 의미와 중요성을 파악하고, 생활 속에서 민주주의를 실천하는 태도를 기른다.</li> <li>• 교과-소프트웨어융합주제 : 민주주의 실천을 위한 전자 투표 프로그램 만들어 보기</li> <li>• 교과-소프트웨어융합학습 목표 : 1) 전자투표 프로그램을 설계할 수 있다. 2) 리스트와 조건문을 활용하여 프로그램을 만들 수 있다. 3) 만든 전자투표 프로그램을 활용하여 투표를 실시할 수 있다.</li> </ul> <p>(엔트리 학습자료 ‘전자투표 프로그램으로 모의선거 실시하기’ 자료 참고)</p>

	지침 3.2 학습자가 친숙함을 느낄 수 있는 시나리오 형태로 문제를 제시하라(off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>[6실03-04] 초등학교 6학년 실과 과목의 ‘자원 관리와 자립’ 단원 내용을 중심으로 환경오염으로 발생하는 생활 주변의 문제점들을 알아보고, 쓰레기 재활용을 통한 환경 보호의 필요성을 안내 후, 분리수거와 관련된 문제를 해결할 수 있는 프로그램 설계하기 (엔트리 학습자료 ‘자동으로 분리수거를 해요’ 자료 참고)</li> </ul>
	지침 3.3 해결안이 하나 이상으로 제시될 수 있는 복합적이고 비구조화, 및 반구조화된 형태의 문제를 설계하라(off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>비구조화 형태의 문제 설계 시 고려할 점 : 1) 문제와 관련된 상황이나 요소가 분명히 정의되어 있지 않은 문제로 설계한다. 2) 문제 해결에 필요한 정보가 충분히 포함되어 있지 않은 문제를 설계한다. 3) 다양한 해답이나 해결 경로를 가지고 있다. 4) 의사결정의 명료화와 논쟁에 초점을 둔다.</li> </ul>
소프트 웨어 학습도구	<b>원리 4. 소프트웨어도구 적합성의 원리</b> 효과적인 수업 구성 및 수업 목표 달성을 위해 적합한 소프트웨어도구를 수업 활동의 목적에 맞게 선정하고 활용한다(on).	
	지침 4.1 계획된 최종 산출물 형태를 효과적으로 구현하기에 적합한 프로그래밍 소프트웨어도구를 선정하고 사전에 설치를 진행하라(on).	
	예시 및 해설	<ul style="list-style-type: none"> <li>프로그래밍 소프트웨어 : 엔트리(<a href="https://playentry.org/">https://playentry.org/</a>), 스크래치(<a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a>)</li> <li>오픈소스 HW : 아두이노, 마이크로비트, 라즈베리파이</li> </ul>

지침 4.2 창의적 문제해결 활동 및 프로젝트 구현을 위한 온라인 학습공간 및 LMS 시스템을 제공하라(on).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 블록형 프로그래밍 소프트웨어 제공 사이트 : 엔트리에서 제공하는 학급 개설 및 공유 기능을 활용하거나, 스크래치에서 제공하는 스튜디오 기능을 활용한다.</li> <li>• 학습관리시스템(LMS) : 학급용 계정을 개설한 후 학습자들의 결과물을 탑재한다(예: 구글 클래스룸, e-학습터, Moodle, Canvas, Edmodo).</li> </ul>
<b>원리 5. 사용 용이성의 원리</b> 교과융합 소프트웨어수업의 학습목표를 효과적으로 달성하기 위해사용 용이성을 가진 학습 도구를 지원한다(on).	
지침 5.1 초보 학습자도 쉽게 사용할 수 있는 직관적 인터페이스의 프로그래밍 소프트웨어 도구를 선정하고 학습자들이 가입할 수 있도록 사전에 안내하라(on).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 엔트리(<a href="https://playentry.org/">https://playentry.org/</a>), 스크래치(<a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a>) 등의 프로그래밍 언어를 쉽고 직관적으로 이해할 수 있는 블록형 프로그래밍 소프트웨어를 활용한다.</li> </ul>
지침 5.2 학습자들이 쉽게 이해하고 사용할 수 있는 소프트웨어기반 사고 지원 도구를 선정하라(on).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 학습 환경과 자원을 고려하여 확보 및 적용이 적절하고 학습자가 쉽게 이해할 수 있는 발산적·수렴적 사고 지원 도구를 선정한다.</li> </ul>

		<ul style="list-style-type: none"> <li>• 마인드마스터, S<sup>3</sup>CPS 등의 창의적 문제해결을 위한 웹기반 사고 지원 도구를 활용한다.</li> </ul>
<b>교과-소프트웨어 융합 학습지원</b>	<b>원리 6. 자료의 명료성의 원리</b> 학습내용과 학습 활동에 대한 안내 자료와 규칙을 알기 쉽고 명료한 형태로 제작하여 제공한다(on/off).	
	지침 6.1 학습 내용과 관련한 검색 키워드 및 참고사이트, 자료 아카이브, 학습자료(코딩 명령어 매뉴얼, 코딩 작성 예시자료)를 사전에 만들어 제공하라(on).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 지정된 시간 내에 효과적으로 주요한 자료를 검색하거나 찾을 수 있도록 검색 사이트, 검색할 수 있는 단어, 참고 사이트 및 주제에 대한 주요 내용을 정리하고 자료의 링크 주소 등을 정리한 자료 아카이브를 온라인 상에 게시하고 활동에 참고할 것을 안내한다.</li> <li>• 프로그램 작성 방법과 사용하는 방법을 순서에 따라 구체적으로 설명한 해설 자료를 제공한다.</li> </ul>
	지침 6.2 수업 진행방식, 학습 과제 및 활동에 대한 분명한 준거(과제 수행 시간, 완성도, 평가 계획)에 대해 안내문을 작성하여 온라인 학습공간에 게시하라(on/off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 공지사항은 항상 열어볼 수 있도록 게시판을 별도로 생성하여 게시하거나 게시판 상위에 노출되도록 하고, 필요시 이미지나 참조링크를 함께 첨부한다.</li> <li>• 필요시 개별 학생에게 공지사항 알림 이메일이 발송되도록 설정한다.</li> </ul>
	지침 6.3 학습자의 자유로운 사고를 촉진하고 학습자간 의견을 부담 없이 공유할 수 있는 열린 학습 분위기 형성을 위해 규칙과 방법을 계획하라(on/off).	

예시 및 해설	<ul style="list-style-type: none"> <li>• 산출물 평가 과정에서 학습자간 자유롭게 토의하고 의견을 제시할 수 있도록 촉진한다.</li> <li>• 실패에 긍정적이고 허용적인 분위기 조성을 위해 규칙과 방법을 제시한다(예: 마인드맵 소프트웨어를 활용하여 친구와 자유롭게 아이디어를 나누기, 나와 다른 아이디어에 대한 장점을 찾아보고 리스트를 만들기, 다른 모듈이 제작한 프로그램의 잘된 점에 대해 한 줄 칭찬 남기기).</li> </ul>
<b>원리 7. 프로그래밍 모듈 구성의 원리</b> 각 학습자에게 역할을 부여하고 학습자 간 동료 프로그래밍 활동을 통해 협력적으로 작업할 수 있는 기회를 제공한다(on/off).	
지침 7.1 컴퓨터를 활용한 프로그래밍 작업 경험이 풍부한 학습자들과 상대적으로 부족한 학습자들을 하나의 모듈으로 구성하라(on/off).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 모든 학습자들이 비슷한 수준의 프로그래밍 경험을 가지고 있는 것이 아니므로, 다양한 수준의 학습자를 하나의 모듈로 구성하여 서로 도움을 주고받으며 경험이 풍부한 학습자가 기술적인 면에서 프로젝트 진행을 보완해 줄 수 있도록 지원한다.</li> <li>• 학습자들의 수준 차이가 클 경우, 이를 보완할 수 있도록 수업 전 소프트웨어 프로그래밍에 대한 충분한 사전 학습을 제공한다.</li> </ul>
지침 7.2 운전자 역할의 학습자와 네비게이터 역할의 학습자가 동시적·협력적으로 작업할 수 있도록 모듈별로 역할을 배정하고 각자의 역할에 대해 설명하라(on/off).	
예시 및 해설	<ul style="list-style-type: none"> <li>○ 동료 프로그래밍(짝 프로그래밍) 적용하기</li> <li>• 두 학습자 또는 팀이 하나의 컴퓨터 앞에서 한 명은 운전자 역할을, 다른 한 명은 네비게이터 역할을 맡아 짝을 이루어 프로그래밍을 수행하도록 안내한다(4인 모듈 시 2인씩 역할 부여).</li> </ul>

		<ul style="list-style-type: none"> <li>운전자 역할 : 키보드와 마우스를 사용하여 타이핑을 하고 코드를 작성한다. 코딩 소프트웨어를 활용한 코드 작성 및 입력 등의 역할을 담당한다.</li> <li>네비게이터 역할 : 프로젝트의 큰 그림에 집중하고 작성된 코드가 논리적인지 확인한다. 스토리보드 스케치, 화면 디자인, 아이디어 도출, 알고리즘 구상, 작성된 코드의 논리 및 오류 확인 등의 역할을 담당한다.</li> </ul>
	지침 7.3 일정 기준을 중심으로 운전자와 네비게이터의 역할을 바꾸어 서로의 작업을 검토하고 보완할 수 있도록 안내하라(on/off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>시간제한이나 해결해야 할 문제의 개수, 일정 수준의 진도, 기타 측정 가능한 기준을 만들어 이를 달성하면 바로 역할을 바꾸어 활동할 수 있도록 안내한다.</li> </ul>
교과-소프트웨어 융합 수업활동	<b>원리 8. 문제이해 지원의 원리</b>	
	실생활과 관련된 학습 자료와 질문프롬프트를 제시함으로써 학습자가 문제를 발견하고 이해할 수 있도록 지원한다(on/off).	
	지침 8.1 수업 내용과 관련된 실생활의 예시 자료(동영상, 애니메이션, 신문기사 등)를 제시하고 학습자의 경험을 공유하도록 안내하라(on/off).	
예시 및 해설	<ul style="list-style-type: none"> <li>학습자가 실생활에서 경험하고 공감할 수 있는 친숙한 문제를 선정하고 참고할 만한 사이트 링크 및 멀티미디어 콘텐츠를 온라인 플랫폼 상에 탑재한다.</li> <li>모둠 또는 개별적으로 자신의 경험과 생각을 온라인 플랫폼 상에 게시하여 서로 댓글을 작성하게 하거나, 발표를 통해 공유한다.</li> </ul>	

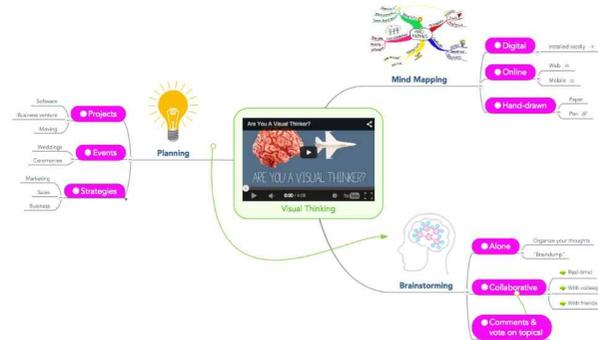
지침 8.2 학습자가 문제를 이해하고 정의할 수 있도록 수업 주제, 해결해야 하는 문제, 해결 목적과 관련한 구체적인 질문을 제시하라(on/off).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 문제에 대해 어떤 의문점이 드나요?</li> <li>• 문제의 원인은 무엇일까요?</li> <li>• 문제와 관련된 이해당사자는 누구일까요?</li> <li>• 무엇이 문제인 걸까요?</li> <li>• 문제를 해결하기 위해 무엇부터 알아보아야 할까요?</li> <li>• 왜 이 문제를 해결해야 할까요?</li> </ul>
지침 8.3 학습자 수준을 고려하여 최종적으로 도출할 수 있는 산출물의 예시를 제시하라(on/off).	
예시 및 해설	<ul style="list-style-type: none"> <li>• 성취기준과 공통 주제, 수업 목표, 학습 내용을 고려하여 범위 내의 내용을 포함한 산출물을 작성할 수 있도록 안내한다.</li> <li>• 산출물 형태는 게임, 애니메이션, 시뮬레이션 등 다양한 형태로 제시될 수 있으나, 학습자 수준을 고려하여 사전에 구현이 가능한 과제를 수행하도록 피드백한다.</li> </ul>
<b>원리 9. 아이디어 생성 지원의 원리</b>	
발산적/수렴적 사고 지원 도구를 활용하여 학습자가 창의적 문제해결을 위한 아이디어를 발견할 수 있도록 지원한다(on).	
지침 9.1 문제에 대한 다양한 해결 아이디어를 도출할 수 있도록 발산적 사고도구(브레인스토밍, 속성열거법, SCAMPER)를 활용하라(on).	
예시 및 해설	<ul style="list-style-type: none"> <li>○ 발산적 사고도구의 종류와 예시</li> <li>• 브레인스토밍 : 일정한 주제에 대하여 자유로운 토론을 통해 아이디어를 생성해 내고자 할 때 사용하는 사고 기법</li> </ul>

		<ul style="list-style-type: none"> <li>• 속성열거법 : 문제상황, 물건, 과제 속성을 열거한 후 이를 변화, 수정함으로써 각 속성 별 아이디어를 도출하는 확장적 사고기법</li> <li>• SCAMPER : S(substitute, 대치하기), C(combine, 결합하기), A(adapt, 응용하기), M(modify - magnify - minify, 수정, 확대, 축소하기), P(put to other use, 새로운 용도), E(eliminate, 제거하기), R(rearrange-reverse, 재배열하기)의 약자로 아이디어 산출을 위해 사고의 출발점 혹은 문제해결의 착안점을 미리 항목별로 써 놓고 하나씩 체크하면서 아이디어를 내고자 하는 체크리스트 법(백연경, 2005)</li> </ul>
<p>지침 9.2 문제에 대한 해결책을 찾을 수 있도록 수렴적 사고도구(HIT, PMI, 평가행렬법)를 활용하라(on).</p>		
예시 및 해설		<ul style="list-style-type: none"> <li>○ 수렴적 사고도구의 종류와 예시</li> <li>○ HIT : 여러 가지 대안들 가운데서 가장 최적의 아이디어에 체크(V)표시를 하는 기법으로 좁혀가야 할 대안의 수가 많을 때 사용함(임철일 외, 2014)</li> <li>○ PMI : 하나의 아이디어의 분석 및 주의를 집중하는 도구로, P는 Plus로 제시된 아이디어의 좋은 점을, M은 Minus로 나쁜 점을, I는 Interesting으로 흥미로운 점을 나타냄(임철일 외, 2014)</li> <li>○ 평가행렬법 : 여러 가지 아이디어를 평가 기준에 따라 각각의 강점과 약점을 파악하여 선택하는데 사용함(임철일 외, 2014)</li> </ul>
<p>지침 9.3 아이디어의 정리와 조직화를 위해 시각화 지원도구(패들릿, 웹기반 브레인스토밍 툴 등)를 활용하라(on).</p>		
예시 및 해설		<ul style="list-style-type: none"> <li>• 패들릿(<a href="https://ko.padlet.com">https://ko.padlet.com</a>) : 모듈별로 게시판을 생성하고 학습내용과 관련된 텍스트나 이미지파일을 업로드하고 공유할 수 있도록 안내한다.</li> </ul>



<패들렛 예시 화면>

- 마인드마이스터(<https://www.mindmeister.com/ko>) : 웹기반 브레인스토밍 도구를 사용하여 다양한 아이디어와 단어들을 효율적으로 정리할 수 있도록 안내한다.



<마인드마이스터 예시 화면>

### 원리 10. 논리화 지원의 원리

학습자가 아이디어의 핵심요소를 파악하고 논리적인 해결안(알고리즘)을 작성할 수 있도록 지원한다(on).

지침 10.1 중요한 아이디어를 선별하여 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호 작용 등을 이해하고 해결안을 표현할 수 있도록 스토리보드 형태의 활동지를 제공하라(on).

예시 및  
해설

- 주요 장면에 대한 시각적 화면 구성과 논리 구성(장면의 텍스트, 블록 코드의 논리적 배열과 순서)이 동시에 이루어지도록 활동지를 이등분하여 구성한다.

➤ 만들고 싶은 프로그램의 장면 모습과 논리 구조를 설계하여 봅시다.

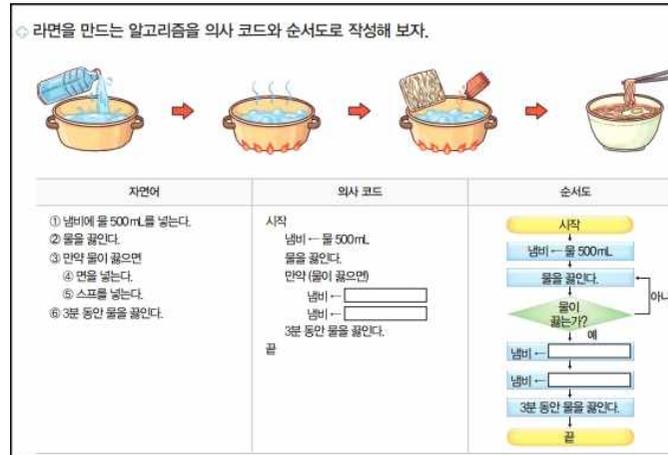
장면 생각하기	논리 생각하기
1)	1)
2)	2)
3)	3)

[활동지 예시]

지침 10.2 의사코드와 순서도의 작성 방법을 안내하고, 작성한 알고리즘이 최종 산출물로 구현될 수 있는지 모듈별 논의를 통해 검토할 수 있도록 안내하라(on).

예시 및  
해설

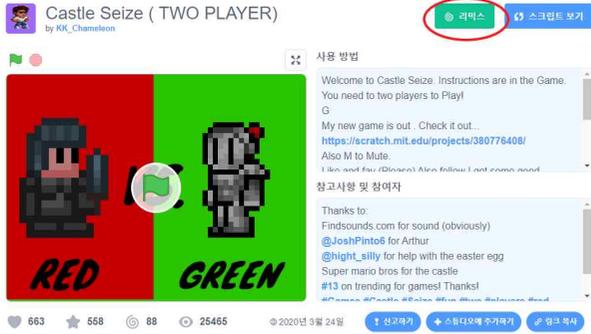
○ 의사코드와 순서도 예시



- 산출물의 형태와 기능을 문장 형태로 표현하게 한다.
- 작성한 문장을 절차적으로 나누어 자연어로 표현하게 한다.
- 나누어진 문장을 의사코드나 순서도의 형태로 표현하게 한다.
- 모듈별로 알고리즘이 프로그래밍을 통해 산출물로 구현 가능한지 검토할 수 있도록 안내한다.

**원리 11. 점진적 문제해결 지원의 원리**

학습자가 점진적인 문제해결 과정을 경험하게 함으로써 산출물을 스스로 계획하고 구현할 수 있도록 지원한다(on).

<p>지침 11.1 학습자가 점진적으로 프로그램을 테스트하고 디버깅할 수 있도록 피드백을 제공하라(on).</p>	
<p>예시 및 해설</p>	<ul style="list-style-type: none"> <li>• 학습자가 프로그램을 실행하면서 코드의 오류를 발견하고 해결하거나 변수나 입력값, 출력값 등의 조절을 통해 최적의 산출물을 도출할 수 있도록 학습자의 진행상태를 관찰하고 필요시 피드백을 제공한다.</li> <li>• 테스트와 디버깅은 문제해결에 대한 실패와 그에 대한 수정 과정이 아닌, 완성품을 만들기 위한 일련의 필수적인 과정임을 안내한다.</li> <li>• 디버깅 : 오류가 발생한 알고리즘의 부분과 발생 원인을 찾아 해결하는 과정</li> </ul>
<p>지침 11.2 학습자의 리믹스 활동을 지원하기 위해 프로그래밍의 구성요소를 일부 제공하거나, 동료나 교수자 및 온라인상의 다른 학습자들이 제작한 소프트웨어의 코드를 열람하고 이를 바탕으로 알고리즘을 발전시킬 수 있도록 안내하라(on).</p>	
<p>예시 및 해설</p>	 <p>&lt;스크래치에서 제공하는 리믹스 기능&gt;</p> <ul style="list-style-type: none"> <li>• 리믹스 : 기존 프로그래밍 코드를 수정하고 재사용하거나 프로그래밍 구성요소의 일부를 제공하여 이를 바탕으로 수정하는 것</li> </ul>

	지침 11.3 컴퓨팅 사고력과 관련한 지식(반복, 연산자, 변수, 알고리즘 등)을 활용하여 점차적으로 복잡하고 어려운 코드를 작성할 수 있도록 난이도를 고려하여 활동순서를 조정하라(on).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 간단한 알고리즘으로 작성할 수 있는 문제 상황을 제시하고 점진적으로 복잡한 문제상황을 제시한다.</li> <li>• 수업 초반에서는 상대적으로 이해하기 쉬운 반복 명령어에 대해 안내하고, 후반에는 조건 명령어를 활용해 보도록 안내한다.</li> </ul>
공유 및 평가	<b>원리 12. 공유 및 평가 지원의 원리</b> 학습자가 학습활동의 산출물을 제시·공유하여 평가 및 피드백을 받을 수 있는 기회를 제공한다(on/off).	
	지침 12.1 모듈 별 발표를 통해 최종산출물을 시연하고 의견을 나눌 수 있는 기회를 제공하라(on/off).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 학습자는 산출물을 만든 이유와 목적, 제작과정, 어려움과 느낀점을 발표하고 시연하며, 이에 대해 모듈 별로 자유롭게 의견을 제시한다. 교수자는 발표 및 논의 과정을 모니터링하고 필요 시 피드백을 제공한다.</li> </ul>
	지침 12.2 학습자의 산출물을 다양한 관점에서 객관적으로 평가할 수 있도록 모듈 별 평가 도구를 제공하라(on).	
	예시 및 해설	<ul style="list-style-type: none"> <li>• 교수자는 온라인 플랫폼 상에서 모듈 별 산출물에 대해 지원적이고 긍정적인 피드백을 중심으로 제공한다.</li> <li>• 다른 모듈에서 발표한 내용에 대해 동료평가의 예시나 잘된 점과 개선할 점, 궁금한 점 등 평가요소가 포함된 평가도구를 제공한다.</li> </ul>

		<ul style="list-style-type: none"> <li>모둠별로 평가 결과를 확인하고 다른 모둠의 질문에 답변하고 개선할 수 있는 점을 작성하도록 안내 한다.</li> </ul>
	<p>지침 12.3 학습자가 전체적인 학습활동을 돌이켜볼 수 있도록 구조화된 성찰일지 및 오픈 포트폴리오 작성을 지원하라(on).</p>	
	<p>예시 및 해설</p>	<ul style="list-style-type: none"> <li>○ 성찰일지 문항 예시 <ul style="list-style-type: none"> <li>문제를 해결하며 만들어낸 결과물이 마음에 드나요? 어떠한 점을 더 개선할 수 있을까요?</li> <li>여러분이 선택한 문제 해결 방법은 무엇이었나요? 그것이 가장 효과적인 방법이라고 생각하나요? 혹시 다른 문제해결 방법을 생각해 볼 수 있을까요?</li> <li>과학(수학)과 소프트웨어를 융합한 다양한 문제들과 해결방법에 대해 더 생각해 볼 수 있을까요?</li> <li>앞으로 문제를 해결해야 할 때 이번 활동으로 배운 것들을 적용할 수 있을까요?</li> </ul> </li> <li>○ 오픈 포트폴리오 내용 요소 예시 <ul style="list-style-type: none"> <li>모둠 소개 및 모둠 역할 분담</li> <li>활동내용(프로젝트 주제, 아이디어 생성 및 논리화 과정)</li> <li>프로젝트 산출물</li> <li>프로젝트를 수행하며 느낀점(잘된 점, 어려웠던 점, 개선이 필요한 점 등)</li> <li>기타 공유하고 싶은 내용(프로젝트 산출물의 조작법, 제작 영상, 설문조사 등)</li> </ul> </li> </ul>

※ 온라인·오프라인 맥락의 수업설계 원리 및 상세지침은 각각 on, off 로 구분하여 제시함

## 나. 절차모형의 단계별 설명

본 절차모형은 본격적인 수업 설계를 위한 각 단계와 세부 절차를 안내한다([그림 IV-18] 참고). 본 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 위한 설계의 전체 과정을 크게 ‘분석’, ‘설계 및 개발’, ‘실행 및 평가’의 세 단계로 분류하여 안내하고 있으며, 총 13개의 절차로 이루어져 있다. 또한 각 설계 단계와 매칭하거나 참고할 수 있는 상세 지침을 함께 표시하였다.

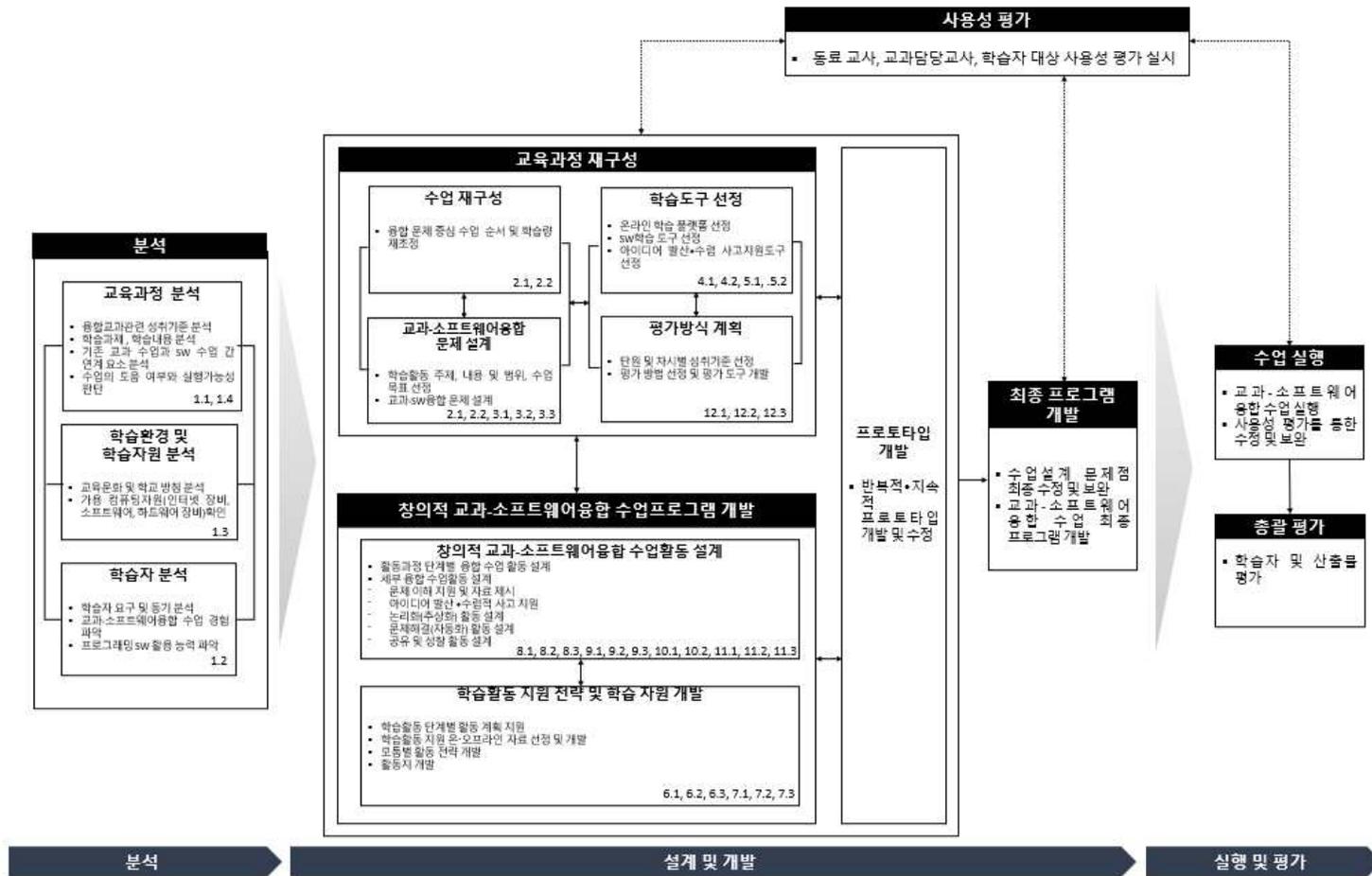
‘분석’단계는 교육과정, 학습 환경 및 학습자원, 학습자 분석 단계로 이루어져 있다. 이 과정에서는 융합을 하고자 하는 교과의 교육과정을 분석하고 교과-소프트웨어융합 수업이 학생들의 학업 성취 및 창의적 문제해결력 향상에 도움이 되는지를 고려하여 수업의 효과성과 실행 가능성을 판단한다. 또한 교과-소프트웨어융합 수업을 진행할 수 있는 학교 내외의 학습 환경과 자원 등 제반 사항들을 확인하고, 학습자의 프로그래밍 소프트웨어 활용 능력 등 학습자 수준을 파악하고 사전 학습의 필요 여부를 판단한다.

‘설계 및 개발’ 단계는 크게 교육과정을 재구성 하는 단계와 창의적 교과-소프트웨어융합 수업프로그램을 설계하는 단계와 최종 프로그램을 개발하는 나뉘어져 있다. 교육과정 재구성 단계는 수업 활동 설계에 앞서 수업의 순서를 재구성하고, 수업에서 다루어질 핵심적인 문제를 설계하는 단계이다. 또한 이러한 과정에서 아이디어의 수렴·발산을 지원하는 온·오프라인 학습도구와 프로그래밍 소프트웨어를 선정하고, 성취기준에 따른 평가방법을 계획한다. 창의적 교과-소프트웨어융합 수업프로그램 설계는 본격적으로 수업을 설계하는 단계이다. 교수자는 활동 단계별로 교과-소프트웨어융합 수업 활동의 전체적인 청사진을 그리고, 세부적으로 문제 및 자료 제시, 아이디어 구성, 논리화 지원 방법 선정, 점진적 문제해결, 공유 및 평가 지원 방법을 설계한다. 이후 학습활동을 지원할 수 있는 전략 및 학습 자원을 개발한다. 모듈별 활동 전략 및 수업에서 제시할 활동지 등을 개발할 수 있다. 특히 교과-소프트웨어융합 수업 특

성 상 교수자와 학습자에게 익숙하지 않은 형태의 수업일 수 있다는 가능성을 고려하여, 실제 수업 실행에 앞서 수업 설계와 관련한 동료 교사들의 의견을 수렴할 필요가 있다. 이에 따라 전반적인 설계 및 개발 단계에 걸쳐 설계한 프로토타입 수업안에 대한 동료교사 및 교과담당교사의 의견을 반영함으로써 반복적·지속적으로 수업 설계의 산출물을 개발 및 수정해 나가는 래피드 프로토타입 방법론을 적용하도록 하였다. 이때 수집된 다양한 의견과 수정사항을 반영하여 교과-소프트웨어융합 수업의 최종 프로그램을 개발해야 한다.

‘실행 및 평가’ 단계는 실제 교과-소프트웨어융합 수업의 실행과 총괄 평가 단계를 포함한다. 실제 수업의 실행과 평가 단계는 수업 설계 이후 본격적인 실행과 운영의 단계에 해당하지만 실제 수업을 통해 얻은 경험적 데이터와 수업 후 사용성 평가 결과를 반영한 수정 및 보완이 지속적으로 수업 설계 단계에 영향을 미친다는 점에서 절차 모형에 포함되는 것으로 하였다.

이에 따라 이하의 각 단계에 대한 설명 부분에서는 ‘분석’ 및 ‘설계 및 개발’ 단계를 중점적으로 설명하였으며, 설명과 함께 각 단계별로 교수자가 실제 수업을 설계 시 참고할 수 있는 예시 및 보충 자료를 함께 제시하였다.



[그림 IV-18] 창의적 문제해결을 위한 교과-소프트웨어 융합 수업설계의 절차모형(최종)

## 1) 분석 단계

### 가) 분석

#### (1) 교육과정 분석

교과-소프트웨어융합 수업 설계는 국어, 수학, 과학 등 기존 정규 교육과정을 바탕으로 소프트웨어 학습 내용의 융합을 통해 이루어진다. 보통 교수자들은 교과서의 내용을 토대로 수업을 설계하지만, 교과-소프트웨어융합 수업은 다양한 교과와 소프트웨어를 융합하여 이루어지기 때문에 수업 설계 초반 교육과정을 분석하는 단계가 필요하다. 이를 위해 기존 교과 수업과 소프트웨어 수업을 연계시킬 요소가 있는지 확인하고, 각 교과의 성취 기준을 분석한다. 성취기준은 교육을 통해 학생들이 성취할 것으로 기대되는 목표로서(김명숙, 2004), 융합할 교과를 중심으로 컴퓨팅 사고력을 적용하여 창의적으로 문제를 해결할 수 있는 융합적인 문제를 염두에 두고 분석해야 한다. 성취기준을 확인한 후 학습과제 및 학습 내용을 분석한다. 학습과제는 수업 목표를 달성하기 위해 필요한 학습 요소들을 확인하고 위계적으로 조직하여 논리적 관련성을 가질 수 있도록 분석되어야 하며, 교과 간 학습 내용이 서로 밀접하게 융합될 수 있어야 한다. 이후 교과-소프트웨어융합 수업을 실행함으로써 학습 목표 달성이 효과적으로 이루어질 수 있는지 여부를 확인한다. 수업의 성취기준과 학습과제 분석을 통해 소프트웨어를 활용하여 교과 간 융합 수업을 수행하는 것이 학습목표 달성에 도움이 된다고 판단된다면, 교과-소프트웨어융합 수업을 설계하고 적용하도록 한다.

#### (2) 학습환경 및 학습자원 분석

학습 환경 및 학습자원 분석은 교과-소프트웨어융합 수업의 설계 및 실행을 위해 갖추어진 제반 여건을 확인하는 단계이다. 교수자는 수업을

설계하며 정규 교과 시간에 교과 융합 수업이 실시되고 수업 중 소프트웨어를 활용하는 학습 활동에 대하여 학교장, 학부모 및 동료 교사의 의견과 같은 학교 내의 문화와 학교 방침을 확인할 수 있다. 또한 와이파이, 유선랜 등의 인터넷 장비, 개인 PC나 노트북, 태블릿 PC와 같은 하드웨어 장비, 수업 진행에 필요한 소프트웨어 등 학습 자원을 확인하고 가용 여부를 분석하여 수업 설계에 반영해야 한다.

### (3) 학습자 분석

학습자 분석 시에는 교과-소프트웨어융합 학습에 대한 학습자의 흥미 및 동기, 요구사항, 개인 PC 및 태블릿 PC의 소유 유무 등을 분석한다. 또한 교과 간 융합 수업, 소프트웨어 활용 수업에 대한 경험을 파악하고, 특히 프로그래밍 소프트웨어에 대한 사전 지식과 활용 능력을 분석할 수 있다. 이러한 학습자의 소프트웨어 활용 능력은 수업 진행 시 학습자의 성취 수준에 많은 영향을 미치므로 선행 학습 여부를 파악하고 필요 시 사전에 기본적인 소프트웨어 활용 수업을 계획하고 실시해야 한다.

## 2) 설계 및 개발 단계

### 가) 교육과정 재구성

#### (1) 수업 재구성

학습자가 수업에서 해결해야 할 문제를 설계하는 단계로 교육과정을 재구성하고 수업의 주제 및 내용 그리고 문제를 설계하는 단계이다. 일반적으로 교육과정 재구성은 교사가 교육과정 및 교과서의 교육내용과 교육활동을 교실 상황에 적절하게 구안하는 것을 의미한다(박일수, 2013). 교과-소프트웨어융합 수업 설계를 위한 교육과정 재구성은 소프트웨어와 타 교과와의 융합을 위해 관련된 교과의 특성, 주제의 연계성,

학습자 수준을 고려하여 학습 내용 및 학습 활동의 순서를 변경하거나, 내용을 추가 및 대체하고, 필요한 경우 생략하거나 축약하는 과정이다. 교육과정을 재구성하면서 연계된 내용을 바탕으로 학습활동의 주제를 설정하고, 이를 하나의 맥락으로 통합할 수 있는 학습 내용 및 범위, 수업의 목표를 설정한 후, 모듈형, 블록 타임형 등 다양한 방법으로 수업 적용 방식을 선정한다.

## (2) 교과-소프트웨어융합 문제설계

교과-소프트웨어융합 문제설계는 교육과정을 재구성하고 수업의 주제와 범위를 설정하여 교과-소프트웨어융합 문제를 구체적으로 설계하고 개발하는 과정이다. 문제 설계 시에는 융합 과목의 소프트웨어 연계 요소를 고려하여 학습 내용과 맥락 및 학습자의 경험과 관련된 실제적이고 비구조화된 문제를 다룬다(<표 IV-29> 참고). 비구조화된 문제는 실생활의 문제와 연계되어 있고, 학습자의 흥미를 유발하며, 학생들에게 프로젝트에 주인의식을 가지고 과제를 해결하기 위한 동기를 부여하고, 아이디어 도출을 위한 모둠 활동이 요구되며, 산출물에 대해 논의과정을 포함하고, 프로젝트 수행을 위해 도구 및 제약사항을 정의하며, 달성하기 어려운 과제라는 특징을 가진다(Capraro, Capraro & Morgan, 2013). 교수자는 융합문제를 설계하면서 학습활동을 통해 학습자가 학습할 수 있는 교과 및 소프트웨어 지식과 학습 과정, 학습 결과를 통한 산출물에 대해 일련의 시나리오를 예상해 볼 수 있다.

<표 IV-29> 학습 맥락과 관련이 있는 비구조화되고 실제적인 문제 설계 시 고려해야 할 기준

구분	기준
문제의 역할	• 문제로부터 학습이 시작되는가?
	• 학습에 필요한 지식과 기능을 충분히 포함할 정도로 포괄적인가?

		<ul style="list-style-type: none"> <li>• 문제에 지식이 사용되는 맥락이나 상황이 제시되어 있는가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 학습자의 역할이 제시되어 있는가?</li> </ul>
비구조성		<ul style="list-style-type: none"> <li>• 문제에 해결에 필요한 일부의 정보만이 포함되어 있는가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 문제해결을 위해 문제를 분석하고, 정보를 찾고, 계획하는 과정이 필요한가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 문제에 대한 다양한 해결책이 존재하는가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 문제해결을 위한 접근 방법이 다양한가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 논쟁이나 토론의 여지가 있는가?</li> </ul>
실제성	일반적 실제성	<ul style="list-style-type: none"> <li>• 실제 사례인가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 일상생활에서 발견될 수 있는 문제인가?</li> </ul>
	물리적 실제성	<ul style="list-style-type: none"> <li>• 현실적인 사물이나 자료를 사용하는가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 문제 해결에 활용되는 사물이나 자료가 다양한가?</li> </ul>
	인지적 실제성	<ul style="list-style-type: none"> <li>• 일상적이고 자연스러운 사고 과정을 반영하는가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 문제 해결에 요구되는 사고 과정이 그 분야의 전문가나 직업인에 의해 사용되는 것인가?</li> </ul>
	관련성	<ul style="list-style-type: none"> <li>• 학습자의 수준에 적절한가?</li> </ul>
		<ul style="list-style-type: none"> <li>• 학습자의 경험과 관련이 있는 문제인가?</li> </ul>
복잡성	<ul style="list-style-type: none"> <li>• 현실과 같이 복잡한 문제인가?</li> </ul>	
	<ul style="list-style-type: none"> <li>• 둘 이상의 문제해결 단계가 필요한가?</li> </ul>	

\* 출처 : 최정임(2004), p49를 재구성함.

### (3) 학습 도구 선정

교과-소프트웨어융합 수업 설계에 필요한 온라인 학습 플랫폼, 소프트웨어를 선정하고, 아이디어 발산·수렴 사고 지원도구를 선정한다. 온라인 학습 플랫폼은 학급 관리와 수업 진행을 지원하는 학습 지원 서비스로서 학급과 수업 관리, 협업, 피드백 활동을 지원한다(<표 IV-30> 참고). 교수자는 학교에서 서비스 이용을 지원하거나 학급 및 수업 관리를 위해 효과적이고 사용이 용이한 학습 및 수업 관리 플랫폼을 선정할 수 있다. 또한 교과-소프트웨어융합 수업에서 활용할 수 있는 협업 플랫폼 및 학습피드백 플랫폼을 선정한다. 협업 플랫폼은 모듈원 간 협업하여 문서를 작성할 수 있는 구글독스(Google Docs)나 포스트잇 형태의 메모를 디지털 캔버스에 자유롭게 작성하며 공동작업을 할 수 있는 패들렛(Padlet) 등의 서비스를 예로 들 수 있다. 학습 피드백 플랫폼으로는 수업 중 실시간으로 퀴즈, 토론을 실시할 수 있는 어플리케이션인 카훗(Kahoot)과 소크라티브(Socrative)를 활용할 수 있으며, 구글 설문지를 통해 평가를 하거나 수업에 대한 학습자의 의견을 수집할 수 있다.

<표 IV-30> 온라인 학습 플랫폼의 종류

구분	플랫폼	설명
학급관리 플랫폼	클래스 123	칭찬카드로 학생들과 소통하고 수업 지원 도구, 학급 보드, 학급 리포트 등 다양한 수업 도구를 갖춘 온라인 학급 경영 서비스 플랫폼
	위두랑	한국교육학술정보원이 만든 온라인 학습 커뮤니티 및 소셜네트워크서비스 플랫폼
	클래스팅	교육용 온라인 학급 커뮤니티 어플리케이션으로 학급 공지 전달, 학생 및 학부모와 콘텐츠 공유 및 관리 등의 기능을 포함
수업관리 플랫폼	구글 G-Suit for Classroom	구글 웹 기반 교육용 공동 작업 및 생산성 도구
	Microsoft	업무에 필요한 교수자 및 학습자, 콘텐츠,

	Teams	도구가 통합된 공동 작업용 허브
	e학습터	17개 시도 통합 초등학교, 중학교 온라인 학습 서비스
협업 플랫폼	구글독스	구글 클라우드 기반 협업 문서 도구
	패들렛	협업을 통한 공동작업을 포스트잇 형태의 메모로 작성할 수 있는 디지털 캔버스
학습피드백 플랫폼	카훗	스마트기기를 활용하여 퀴즈, 토론, 설문 등을 실시간으로 진행할 수 있는 어플리케이션
	소크라티브	스마트기기를 활용하여 퀴즈, 토론, 설문 등을 실시간으로 진행할 수 있는 어플리케이션이며, 클래스 단위로 방을 생성하여 운영 가능
	구글 설문지	구글 웹기반 설문 관리 소프트웨어

교사는 수업 설계 단계에서 수업의 주제와 내용을 다루기에 적합하고 사용하기 용이한 소프트웨어 학습 도구를 선정할 필요가 있다. 소프트웨어를 활용한 프로그래밍을 할 수 있는 학습도구는 대표적으로 MIT 미디어랩에서 제작하고 보급한 교육용 프로그래밍 언어인 스크래치(Scratch)와 2014년 국내 기업이 제작한 소프트웨어 교육 플랫폼인 엔트리(Entry)가 있다. 스크래치는 그래픽, 소리, 효과 등 멀티미디어 처리 기능이 뛰어나고, 세계 각국의 학습자들이 공유하는 제작물과 소스를 활용할 수 있는 장점이 있다. 엔트리는 학급 기능을 제공하여 학급 단위로 학교 수업 시간에 활용하는 것에 적합하고, 엔트리파이썬 기능을 제공하여 블록언어와 텍스트 언어의 변환 및 작성이 가능해 수업의 내용 및 난이도를 심화시킬 수 있다는 특징을 가진다.

또한 소프트웨어에 대한 교육용 동영상과 튜토리얼을 제공하는 소프트웨어 학습지원 웹사이트를 사전 학습에 활용하거나 수업 진행 시 참고할 수 있도록 안내할 수 있다. 소프트웨어 학습 도구 및 교육용 웹사이트를 정리하면 아래 <표 IV-31>와 같다.

<표 IV-31> 소프트웨어학습도구 및 소프트웨어학습지원 웹사이트

구분	서비스	설명
소프트웨어 학습도구	스크래치	MIT 미디어랩에서 제작한 오픈소스 교육용 프로그래밍 언어
	엔트리	국내 기업 네이버에서 제공하는 소프트웨어 교육 플랫폼
소프트웨어 학습지원 웹사이트	code.org	미국 내 비영리 조직이 운영하는 소프트웨어프로그래밍 교육 웹사이트로 전 세대의 학습자가 참여할 수 있는 학습 프로그램 및 다양한 튜토리얼 제공
	이숲	EBS에서 제공하는 교육용 무료 소프트웨어교육 온라인 플랫폼으로 다양한 강좌 및 교육 프로그램 제공
	소프트웨어야놀자	소프트웨어, 인공지능 및 데이터 관련 교육용 동영상, 학습자 및 교수자 대상의 교육 프로그램, 다양한 소프트웨어 관련 뉴스 제공

교과-소프트웨어융합 수업 진행을 위해 수업에 적합하며 학습자들이 이해하기 쉽고 사용이 편리한 아이디어 발산과 수렴을 위한 사고지원 도구를 선정해야 한다. 발산적 사고 지원 도구는 아이디어의 생성을 촉진하는 도구로 브레인스토밍, 속성열거법, 강제결부법, 형태분석법, SCAMPER 등이 있으며, 수렴적 사고도구는 생성된 아이디어를 정리하여 적합한 해결책을 찾는 것을 지원하는 도구로 히트, PMI, 평가행렬법, ALU, 쌍비교분석법 등이 있다.

#### (4) 평가 방식 계획

교과-소프트웨어융합 수업을 통한 창의적 문제해결력 향상을 위해 적절한 평가 방식을 선정한다. 일반적으로 소프트웨어 수업의 성취기준 달성을 위한 객관적인 평가 방법 중 하나는 프로그래밍 스크립트가 적절히 작성되었는지를 확인하는 것이다. 이를 위해서 알고리즘이 적절하게 작성되었는지, 프로그래밍 명령어 및 조건, 반복문 등이 적절하게 사용되었는지, 또한 창의적으로 문제를 해결하였는지 등을 평가해야 한다. 또한 descatch, KOI Study, Coding is fun, Dovelet, Code up 등 알고리즘을 분석하고 평가할 수 있는 웹사이트를 통해 학습자가 작성한 코드의 적합성을 측정할 수 있다. 프로그래밍 코드 평가 항목은 다음 <표 IV-32>와 같다.

<표 IV-32> 프로그래밍(Scratch) 코드 평가 항목

영역	세부항목	설명
기본 개념	Command	- 기본 명령문 및 블록 조합
	Parameter	- Value를 리턴하는 argument의 원리
	Variable	- Value, reference, scope의 원리
	Trigger	- Event 생성 및 제어 원리
	Conditional Statement	- 조건문을 통한 분기제어 원리
	Iteration	- 반복문을 통한 구문의 반복제어 원리
	Data Type	- Boolean, number, string의 제어
	Input & Output	- 키보드, 마우스 등을 통한 데이터 입출력 및 상호작용 제어원리
	Data Structure	- List를 이용한 데이터 제어 원리
	Concurrency	- Event handler를 이용한 multi-threading 생성 및 제어
Procedures	- Make & Block 기능을 이용한 subroutine	

		의 생성, 함수의 재사용, Parameter 제어
복합 개념	Algorithms	- 문제를 개선하기 위한 전략의 설계
	Divide & Conquer	- 문제를 작은 인스턴스로 나누어 top-down 또는 bottom-up 방식으로 해결해나가는 능력
	Interaction Design	- Dynamic data input, sprite 속성 정보를 이용한 다양한 상호작용의 설계 및 구현 능력
콘텐츠 영역	창의성	- 구현 가능한 콘텐츠 스토리의 완성도 - 풍부한 스토리 전개 및 복수의 시나리오 제공 등
	심미성	- 멀티미디어 및 오브젝트의 커스터마이징을 통한 심미적 효과의 존재 여부
	기능성	- 콘텐츠의 기능적 요인 존재여부
	완성도	- 스토리 전개의 다양한 패스 존재여부 - 사용자 선택이 가능한 옵션의 존재여부 - 스토리 구현의 완성도 등

\* 출처 : 손원성(2016), p. 3~4 재구성함.

한편 교과-소프트웨어융합 수업의 평가방식은 성취기준을 달성할 수 있는 객관적인 평가 방법을 기본 바탕으로, 지식과 개념 위주의 전통적인 평가 방안에서 나아가 창의적 문제해결이라는 핵심 역량을 평가하기 위한 지원 방안으로서 선정해야 한다. 이는 일회적 차원에서 측정평가될 수 있는 영역이 아니므로, 발달 과정을 지속적으로 관찰평가하고 학습 수행 중에 이루어지는 형성평가를 통해 지속적이고 정확한 피드백을 제공해 주는 것이 필요하다(김경훈 외, 2015). 더불어 교과-소프트웨어융합 수업은 학습자의 프로젝트 수행과 산출물 제작 과정이 중점적으로 이루어지므로, 이에 대해 중점적으로 판단하는 수행평가를 통해 이루어져야 한다(김성욱, 2016).

이러한 내용을 바탕으로 김수환, 송상수, 임선명, 김지현(2017)은 소프

트웨어교육에서 학생들이 직접 만든 산출물을 토대로 한 포트폴리오 평가를 지원해야 한다고 하였으며, 다음 수준으로의 학습을 촉진할 수 있는 피드백이 가능한 평가도구가 필요함을 언급하였다. 교과-소프트웨어 융합 수업에서는 최종 산출물을 완성하기까지 각 차시마다 학습자의 학습 활동 산출물이 과정 및 단계별로 온라인상에 누적되므로 이를 종합적으로 정리하고 평가할 수 있는 포트폴리오 평가가 적절하다. 포트폴리오 평가를 통해 프로그래밍으로 해결 가능한 문제를 선정하고 적합한 자료를 수집하였는지, 창의적 문제해결을 위한 다양한 아이디어를 발산하고 수렴하였는지, 문제 해결을 위해 적절한 추상화 과정을 거쳐 알고리즘을 설계하고 프로그램으로 구현하였는지 등을 종합적으로 평가해야 한다.

## 나) 창의적 교과-소프트웨어융합 수업프로그램 개발

### (1) 창의적 교과-소프트웨어융합 수업활동 설계

교과-소프트웨어융합 문제를 바탕으로 본격적인 창의적 교과-소프트웨어융합 수업활동을 설계한다. 활동 설계는 개념모형에서 제시한 수업설계의 주요 구성요소 및 진행 단계를 고려하여 설계한다. 이하에서는 교과-소프트웨어융합 수업활동 설계 시 참고할 수 있는 주요 내용을 제시하도록 한다.

#### (가) 문제 이해 지원 및 자료 제시

교과-소프트웨어융합 수업은 문제를 이해하는 단계부터 시작된다. 수업 설계 시 교수자는 문제를 제시하고, 학습자가 문제를 발견 및 이해하도록 지원해야 한다. 이를 위해 주제와 관련한 동영상, 신문기사, 교과서의 사례를 제시하여 학습자의 흥미와 동기를 유발한 후, 실생활 속에서 쉽게 찾아볼 수 있는 문제 상황을 제시하도록 한다. 이 때 학습자가 문제를 분석하고 이해할 수 있도록 관련 정보 및 자료를 제공하고, 해결방

안을 탐색해 볼 수 있도록 문제에 대한 의문점, 문제의 원인, 해결을 위해 우선적으로 고려해야 할 점, 핵심적인 문제점 등의 질문을 제시할 수 있다. 예컨대, 환경오염에 대한 주제를 선정하고 세계 기후 변화에 대한 데이터를 제공해 분석하도록 한 후, 무엇이 문제이고, 원인과 해결해야 하는 이유 등을 질문하여 학습자의 문제에 대한 이해도를 높이고 문제를 정의할 수 있도록 수업을 설계할 수 있다. 교수자가 제공하는 학습자료 이외에도 학습자가 스스로 문제 상황에 대한 자료를 찾고 조직화 할 수 있도록 지원한다. 이 때 학생들이 수집한 데이터는 교수자가 실시간으로 확인할 수 있도록 구글 드라이브 등의 클라우드 서비스에 업로드 하도록 안내한다. 또한 수업에서 다룰 소프트웨어 학습 도구를 안내하고 이를 활용하여 도출할 수 있는 산출물의 형태와 예시를 구체적으로 제시하여 학습자들이 산출물 제작 과정에서 참고할 수 있도록 한다.

#### (나) 아이디어 발산·수렴적 사고 지원

학습자들이 아이디어를 발산하고 수렴할 수 있도록 사고를 지원한다. 창의적 문제해결에는 확산적 사고와 수렴적 사고와 같은 고등사고력이 필수적으로 요구된다(구양미 외, 2006). 확산적 사고는 창의적 문제해결 과정에서 문제를 발견하여 정의하고 새롭고 독창적인 해결책 또는 대안을 찾아보는데 기여하는 사고기능으로, 새로운 방식으로 결합하거나, 새롭고, 신기하고, 독창적인 산출물을 내는 과정이다(조연순 외, 2008). 창의적 사고는 많은 양의 아이디어 찾기, 다양하고 독창적인 아이디어 찾기, 아이디어를 확대 또는 심화시켜 갈 수 있는 세부적이고 구체적인 설명하기 등을 포함한다(Treffinger, Isaksen, Dorval, 2000).

이러한 창의적 사고는 수렴적 사고와 밀접하게 결합되어 있다(Jonassen, 1997). 수렴적 사고는 논리적이고 체계적으로 사고하는 기능으로, 문제를 명료화하고, 가능한 방안들의 타당성을 평가함으로써 문제의 해결책을 구체화 하는데 필요로 하는 사고이다(구양미 외, 2006; 조연순 외, 2008). 수렴적 사고는 비판적 사고라 말하기도 하며, 가능성을 조

직화하고 분석하기, 유망해 보이는 가능성들을 다듬고 개발하기, 대안들의 우선순위 매기기, 그리고 어떤 대안을 선택하거나 의사를 결정하기를 포함한다(Treffinger, Isaksen, Dorval, 2000)

효과적으로 문제를 해결하기 위해서는 아이디어를 생성하고 수렴하는 두 가지 활동을 모두 수행해야 한다. 이를 위해서는 앞서 선정한 사고 지원 도구를 활용하도록 한다. 예를 들어 아이디어 발산 활동을 위해 온라인 협업 도구인 패들렛(Padlet)을 활용하여 브레인스토밍 기법을 적용할 수 있다. 패들렛을 통해 자유롭게 아이디어를 작성하고 디지털 보드에 게시하도록 하며, 이 때 교수자는 이미지, 동영상, 신문기사 및 온라인 사전과 같은 텍스트 링크 등을 추가할 수 있도록 안내하여 학습자들이 관련 자료를 다양하게 접하면서 활발히 아이디어를 생성하도록 촉진한다. 이후 아이디어를 조직화하고 선택하여 수업에 활용할 수 있도록 아이디어 수렴 지원 사고 도구를 적용한다. 예컨대 교수자는 PMI 기법을 적용하여 패들렛에 작성한 아이디어들에 추천이나 좋은 점, 나쁜 점, 흥미로운 점 등에 대한 의견을 남길 것을 안내하고, 학습자들이 아이디어를 구조화시키고 우선순위를 매겨 선택할 수 있도록 촉진할 안내할 수 있다.

#### (다) 논리화(추상화) 활동 설계

ISTE-CSTA-NSF(2011) 등 미국의 정보교육 관련 단체와 국내 교육부(2015)의 2015 개정 교육과정 등에서 제시한 추상화의 정의를 살펴보면, 다음과 같은 개념들을 포함하고 있다(김동만, 이태욱, 2019). 1) 문제 해결에 필요하다. 2) 인간의 사고과정이다. 3) 복잡한 문제를 분해한다. 4) 불필요한 요소를 제거한다. 5) 상세함을 숨긴다. 6) 복잡함을 단순화한다. 7) 필요한 요소를 추출한다. 8) 단순한 형태로 만든다. 즉, 추상화는 특정 세부 사항을 생략해 좀 더 일반적인 문제에 맞는 해결법을 찾는 방법으로, 문제 해결 과정을 더 손쉽게 만들기 위해 세부적인 것을 줄이는 과정이다(Krauss & Prottsman, 2017).

따라서 교과-소프트웨어융합 수업 설계 시 실생활의 문제를 분석하고, 핵심요소를 추출하여 해결 가능한 형태로 만드는 추상화 과정을 경험하도록 하는 것이 중요하다(김경훈 외, 2015). 추상화에 대한 수업을 설계하기 위해서는 실생활의 친숙한 문제 상황을 제시하고, 문제 상황에 제시된 다양한 요소들을 분석하여 목표 상태에 도달하기 위해 필수적인 요소를 찾도록 해야 한다. 그리고 문제의 해결과정을 알고리즘으로 구상하고 논리적인 해법을 탐색하여 절차적이고 논리적으로 명확히 표현할 수 있도록 학습 활동을 구성해야 한다. 예를 들어 초등학교 과학과 연계하여 물 부족과 물의 순환에 대한 애니메이션을 제작할 때, 세부적인 과정을 생략하고 중요한 내용만을 반영하여 만들어가는 과정을 추상화로 볼 수 있다. 이 때 중요 내용 선정하고 이를 절차적으로 구성할 수 있도록 개요를 짜거나 장면별로 스토리보드를 작성할 수 있다. 활동 후에는 학습자들 간 추상화에 대한 경험을 공유하고 토론하는 시간을 제공할 수 있다. 다음은 추상화에 대해 참고할 수 있는 질문 목록이다(<표 IV-33> 참고).

<표 IV-33> 학습자의 추상화 활동에 대한 질문 목록

질문 목록
<ul style="list-style-type: none"> <li>• 활동을 할 때 무엇을 추상화할지 파악하기 어려울 때가 있었는가?</li> <li>• 추상화에서 어떤 면이 이 과제를 더 쉽게 만들었나?</li> <li>• 템플릿/스토리보드/개요 등 추상화 산출물을 만든 것처럼 이번 과제에서 정보를 추상화 할 수 있는 다른 부분은 무엇이 있나?</li> <li>• 정보를 단순화하여 영구적으로 추상화한 다른 사례는 무엇이 있나?</li> </ul>

\* Krauss & Prottzman(2017), p. 101~102 재구성함.

#### (라) 문제해결(자동화) 활동 설계

자동화는 컴퓨팅 시스템이 수행할 수 있는 형태로 해결책을 나타내는 과정으로, 추상화 과정을 통해 도출된 문제 해결 모델은 프로그래밍을

통해 자동화된다(류미영, 한선관, 2018; CSTA & ISTE, 2011). 교과-소프트웨어융합 수업에서는 문제해결 단계에서 자동화 과정이 진행되며, 수업 활동 설계 시 프로그래밍 언어의 개발 환경 및 특성을 이해하고 입력과 출력, 변수와 연산, 실행 흐름 제어를 위한 제어 구조 등 프로그래밍의 기본 개념과 원리를 문제 해결에 적용하도록 하는 것이 중요하다(김경훈 외, 2015). 특히 실생활의 문제 상황을 해결하는 프로젝트 수행을 통한 창의적 문제해결력을 신장하는 데 초점을 두고, 프로그래밍 언어의 지식과 기능만을 습득하기 위한 수업에 치중되지 않도록 유의한다.

교수자는 사전에 프로그래밍 소프트웨어 수업을 진행하여 학습자가 기본적인 지식을 갖추고 수업에 참여하도록 교육과정을 조정하거나, 수업 초기 이미 작성된 프로그램 코드를 동일하게 만들어보고 부분적으로 수정하는 활동을 통해 프로그래밍의 기본 개념과 원리를 습득하도록 한다. 학습 활동 중에는 학습자가 자동화 활동을 수행하며 점진적으로 프로그램을 테스트하고 디버깅할 수 있는 기회를 제공하고, 스크립트의 일부를 제공하여 프로그래밍의 구성요소를 익히고 리믹스할 수 있도록 지원한다. 다음은 수업 후 자동화를 통한 문제해결 과정에 대해 학습자들과 토의하고 의견을 나누어 볼 수 있는 질문 목록이다(<표 IV-34> 참고).

<표 IV-34> 학습자의 문제해결(자동화) 활동에 대한 질문 목록

---

질문 목록

---

- 자동화 과제에서 가장 어려웠던 부분은 무엇인가?
  - 더 나은 자동화를 위해 해결안(알고리즘)에 문제가 있는지 확인하고 수정할 기회가 있었나?
  - 다른 작업을 위한 해결안(알고리즘)을 만들 수 있나?
  - 해결안(알고리즘)과 자동화의 관계는 무엇이라고 생각하나?
- 

\* Krauss & Prottzman(2017), p. 112 재구성함.

(마) 공유 및 성찰 활동 설계

산출물 공유 단계에서는 교과-소프트웨어융합 학습 활동의 결과를 보고서, 동영상, 파워포인트 등과 같은 형식으로 정리하고 이를 온라인 상의 학급 게시판이나 클라우드에 공유 및 시연하도록 한다. 교수자, 모둠원, 모둠원 외의 학급 구성원들은 산출물에 대해 강점, 약점, 개선점 등을 중심으로 평가 및 피드백을 하고, 학습자는 이를 반영하여 산출물을 다시 수정 및 보완하는 작업을 수행하도록 한다.

이후 교수자는 학습활동 전반에 대한 성찰 활동이 이루어지도록 한다. 이를 위해 구글 설문지와 같은 온라인 문서 도구를 활용하여 학습자가 성찰할 수 있는 질문을 제시하고, 이에 대해 답변할 수 있도록 안내하거나, 수업활동의 단계별 산출물을 정리하고 프로젝트를 수행하며 느낀점 등을 포함한 오픈 포트폴리오를 제작하게 할 수 있다. 오픈 포트폴리오는 공개적인 네트워크로 확산 및 공헌 가능한 포트폴리오로, 학습자는 오픈 포트폴리오를 통해 자신 또는 모듬의 활동을 차시별, 단계별로 기록할 수 있고, 다른 학습자들의 평가 및 피드백도 받을 수 있다. 오픈 포트폴리오는 Google Sites 등 온라인 웹페이지 제작 도구를 활용할 수 있으며, 팀원소개, 팀 내 역할, 산출물 제작과정, 문제해결 과정, 최종산출물, 프로젝트 성찰내용(잘된 점, 어려움을 겪은 점과 이를 해결한 점, 개선이 필요한 점 등) 등을 이미지, 동영상을 첨부하여 전달력 있게 정리할 수 있도록 안내한다.

## (2) 학습활동 지원전략 및 학습자원 개발

학습활동의 지원전략을 설계하고, 학습 자원을 개발한다. 학습활동 단계별 활동 계획 시에는 교과-소프트웨어융합 문제 상황에 대해 어떤 아이디어를 토대로 어떤 방법을 통해 해결할 것인지를 모듬별로 토의하고 문제 이해, 아이디어 구성, 논리화, 문제해결의 단계별 과정에 대한 계획서를 작성하도록 양식을 제공할 수 있다. 또한 단계마다 모듬 단위로 수행할 수 있는 활동을 분류하고 모듬원의 의견을 바탕으로 아이디어를 선정하거나, 논의를 통해 알고리즘을 개발하고, 동료 프로그래밍 활동을 하

는 등의 모듈별 활동 전략을 개발하도록 한다. 이 때 모듈별 학습활동을 보조할 수 있는 활동지를 제공하거나, 온라인 협업 소프트웨어에 템플릿 형태로 게시하여 학습 활동을 지원할 수 있다.

더불어 학습활동에 활용할 수 있는 온·오프라인 자료를 탐색하고 적절한 자료를 선정하도록 한다. 자료는 온·오프라인 상의 도서, 신문기사, 이미지, 동영상, 공개교육자료(OER) 등을 활용할 수 있으며, 학습 활동 중 참고할 수 있는 자료를 아카이브의 형태로 웹에 게시하거나, 스크랩하여 학습자들에게 배포할 수 있다.

### (3) 프로토타입 개발, 사용성 평가 및 최종 프로그램 개발

분석과 설계, 개발 단계를 거쳐 설계된 수업 프로그램의 프로토타입을 반복적인 사용성 평가를 통해 빠르게 검토하여 수정 및 보완하도록 한다. 동료 교사 및 교과담당 교사, 학습자들을 통해 사용성 평가를 진행할 수 있으며, 수업의 실행 단계에서도 실시할 수 있다. 동료 교사의 피드백 시에는 각 단계의 수업 설계 결과물을 동료 교사에게 설명하여 목표를 달성하기에 적합하고 실행에 무리가 없을지에 대한 피드백을 받도록 한다. 또한 소프트웨어와 타 교과 간의 융합 수업 설계를 위해 필요 시 해당 교과를 담당하고 있는 교사에게 피드백을 요청하고 융합 수업을 위한 교과 내용으로서 적절한지를 묻고 피드백 받을 수 있다. 학습자 사용성 평가를 위해서는 일부 학습자를 대상으로 수업 실행 전후로 수업의 내용과 난이도에 대해 묻고 반응을 확인한 후 이를 설계에 반영하도록 한다.

이와 같이 반복적·지속적인 프로토타입 개발과 수정을 통해 수업 설계의 문제점을 보완하고 최종적인 교과-소프트웨어융합 수업 프로그램을 개발할 수 있다.

### 3) 실행 및 평가 단계

교과-소프트웨어융합 수업을 실행하고 설계한 수업을 평가하여 이후

에 진행되는 수업 설계활동에서 수정 및 보완이 이루어져야 할 부분들을 확인한다. 수업 실행 과정에서는 학습자들이 문제 이해 및 자료수집, 아이디어 생성, 논리화, 문제해결, 공유 및 평가 활동을 원활하게 진행할 수 있도록 지원한다. 수업의 실행 과정에서 학생들의 반응과 성취도를 살피고, 학생들이 어려움을 느끼거나 활동하는 데 무리가 있는 것으로 판단되는 부분을 고려하여 차후 수업 설계에 수정사항으로 반영하도록 한다. 학습자가 문제 상황을 잘 이해하는지, 아이디어의 발산과 수렴이 원활이 이루어지는지, 사전 지식이 부족하여 프로그래밍 활동을 하는 데에 어려움을 겪는지, 학습 플랫폼이나 온라인 협업 도구가 학습 진행에 효율적으로 활용되는지 등 수업 전반에 대한 사항을 평가하고 개선점을 도출한 뒤 수정하도록 한다. 예를 들어 학생들이 프로그래밍 소프트웨어 운용에 어려움을 느낀다면, 수업 전 기초적인 지식을 충분히 전달할 수 있는 차시 또는 모듈을 추가하거나, 프로그래밍 강좌 동영상을 제시하여 미리 학습할 수 있도록 수업을 설계할 수 있다. 평가 과정을 통해 도출된 수정사항은 지속적·반복적으로 환류되어 다음 수업 설계 시 반영되도록 한다.

## V. 논의 및 결론

### 1. 논의

본 연구에서는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계를 지원하기 위한 개념 모형, 절차 모형, 수업설계 원리를 개발하고자 하였다. 개발한 모형과 원리에 대해 전문가 검토와 사용성 평가를 통한 내적 타당성을 검토하고, 수업 적용을 통한 외적 타당성 검토를 통해 교수자와 학습자의 반응을 확인하였다. 이러한 타당화 결과를 반영하여 최종 수업설계 모형과 원리를 개발하였다. 이상의 연구 과정을 통해 최종 결과로 도출된 교과-소프트웨어융합 수업설계 모형과 원리에 대한 시사점을 논의하면 다음과 같다.

#### 가. 교과-소프트웨어융합 수업 설계모형

본 연구에서는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계를 지원하기 위한 개념모형, 절차모형, 수업설계 원리를 개발하였다. 교수자는 개념모형을 통해 본 수업 설계의 의미와 핵심 설계요소를 파악하고, 절차모형을 통해 수업설계의 전체 과정과 절차, 각 단계마다 수행해야 하는 세부 활동을 확인할 수 있다(김성욱, 2016). 설계 원리는 창의적 문제해결, 컴퓨팅 사고력과 소프트웨어 교육, 융합 교육에 대한 연구들을 통합하여 개발되었으며, 구성요소별로 범주화하여 모형의 이론적 기저로서 기술되었다.

교과-소프트웨어융합 수업의 개념모형은 ‘교과-소프트웨어융합 교육과정’, ‘교과-소프트웨어융합 수업활동’, ‘교과-소프트웨어융합 학습지원’, ‘공유 및 평가’, ‘교과-소프트웨어융합 학습도구’의 다섯 가지 구성요소로 제시하였다. 수업 설계는 ‘교과-소프트웨어융합 수업 활동’을 중심으로 이루어진다. 이러한 설계활동의 전후로 ‘교과-소프트웨어융합 교육과정’ 구성을 통해 전반적인 활동의 방향성을 설정하고, ‘공유 및 평가’를 통해

전체 활동을 마무리한다. 세 구성요소는 ‘학습환경 요인’에 의해 영향을 받는다. ‘학습환경 요인’에 영향을 받는 ‘교과-소프트웨어융합 교육과정’, ‘교과-소프트웨어융합 수업 활동’, ‘교과-소프트웨어융합 학습지원’, ‘공유 및 평가’의 네 가지 구성 요소는 ‘교과-소프트웨어융합 학습 도구’의 지원을 받게 된다. 교수자는 ‘교과-소프트웨어융합 수업 활동’을 중심으로 수업을 설계해야 한다. ‘교과-소프트웨어융합 교육 과정’은 수업 설계를 위해 교육과정과 성취기준, 교과내용을 분석 및 재조직하는 것을 의미하며, 수업 활동 설계 전 ‘교과-소프트웨어융합 교육 과정’ 구성을 통해 전 반적인 활동의 방향성을 설계할 수 있다. ‘교과-소프트웨어융합 수업 활동’은 문제 이해, 아이디어 생성, 논리화, 문제해결 수업 활동 설계를 위한 요소이며, 학습 자료 제작 및 모듈 구성 등 수업 활동을 지원하는 ‘교과-소프트웨어융합 수업 지원’과 상호지원관계를 맺는다. 이러한 수업 활동 및 학습지원 요소는 수업 산출물의 시연, 공유, 평가를 포함하는 ‘공유 및 평가’요소로 마무리된다. 네 구성요소는 ‘학습 환경 요인’에 영향을 받으며, ‘교과-소프트웨어융합 학습 도구’의 지원을 받는다. ‘학습 환경 요인’은 교과-소프트웨어융합 수업에 대한 학교와 동료 교사들의 지원과 협조 등의 사회적 지원, HW 및 소프트웨어, 네트워크 인프라 등의 물리적 지원으로 구성된다. ‘교과-소프트웨어융합 학습 도구’는 학습 환경 요인에 포함되고, 프로그래밍 자원, HW자원, 수렴·발산 사고지원 도구 및 이러닝 플랫폼 등의 학습 자원과 도구를 의미하며 상기 네 구성 요소를 지원한다. ‘교과-소프트웨어융합 교육과정’, ‘교과-소프트웨어융합 수업 활동’, ‘교과-소프트웨어융합 학습지원’, ‘공유 및 평가’ 요소는 상호 영향을 주고받는 관계이며 이는 수업 설계 시 네 구성요소를 유기적으로 고려해야 함을 의미한다.

절차모형은 교과-소프트웨어융합 수업을 위한 수업 설계의 전체적인 과정을 ‘분석’, ‘설계 및 개발’, ‘실행 및 평가’의 큰 흐름으로 구분하고, ‘교육과정 분석’, ‘학습 환경 및 학습자원 분석’, ‘학습자 분석’, ‘수업재구성’, ‘교과-소프트웨어융합 문제 설계’, ‘학습도구 선정’, ‘평가방식 계획’, ‘창의적 교과-소프트웨어융합 수업활동 설계’, ‘학습활동 지원 및 학습 자

원 개발’, ‘최종 프로그램 개발’, ‘수업 실행’, ‘총괄 평가’ 순으로 절차를 나타내는 수업 설계 단계와 세부 활동을 제시하고 있다. 이러한 수업설계 개념모형, 절차모형 및 수업설계 원리를 논의하자면 다음과 같다.

첫째, 교과-소프트웨어융합 수업설계를 위해 우선적으로 학습자 및 학습 환경과 학습자원에 대한 분석이 이루어져야 할 필요가 있다. 본 연구에서는 교수자의 입장에서 교과-소프트웨어융합 수업을 설계하기 위해 각 단계마다 고려해야 할 활동들은 무엇인지에 대해 확인하도록 함으로써 수업 설계의 전체적인 과정과 절차를 안내하였다. 특히 교과-소프트웨어융합 수업은 코딩 소프트웨어를 비롯하여 다양한 학습지원 소프트웨어를 활용하는 수업이므로 학습자 분석을 통해 학습자가 어느 정도의 관련 지식을 가지고 있고 자유롭게 활용이 가능한지, 그리고 수업에 대해 어느 정도의 흥미와 동기를 가지고 있으며, 어떠한 요구가 있는지를 사전에 분석해야 할 필요가 있다. 또한 학습 환경 및 학습자원에 대한 분석을 통해 가용할 수 있는 소프트웨어 및 하드웨어 인프라를 확인하고 교과-소프트웨어융합 수업에 대한 학교의 방침 및 문화를 고려한 수업설계가 필요하다. 이러한 관점에서 수업설계의 처음부터 마지막까지의 과정을 체계적으로 다룬 본 모형은 교수자의 입장에서 가장 먼저 분석 단계를 거쳐야 한다는 점을 강조하고 있다.

둘째, 교육과정에 대한 종합적 분석과 이에 따른 재구성을 통해 교과 공통주제를 도출하고 융합적 수업활동을 할 수 있는 ‘교과-소프트웨어융합 문제’를 설계해야 한다. 교사는 융합하려는 교과의 학업성취기준을 파악하고 분류하여 이를 중심으로 주제를 선정하고 교육과정을 재구성한다. 교육과정 재구성 시에는 학습내용, 학습 환경, 학습자 수준과 흥미, 학생들의 이해도, 학습 차시 등을 고려하여 새로운 내용을 추가하고, 중요하지 않거나 중복된 내용을 생략하며, 어려운 내용을 심화하거나 축소하는 등의 재조직 과정이 필요하다(서경혜, 2009; 이영기, 손창호, 2016; 임화영, 윤혜경, 2019; 지경옥, 2017). 또한 교과 특성, 주제의 연계성을 고려하여 융합 주제를 중심으로 하나의 맥락을 구성하여 통합하는 것이 중요하다. 교과-소프트웨어융합 주제는 실제 생활의 문제를 선정하고,

학습자가 친숙함을 느낄 수 있는 시나리오 형태로 제시하며, 해결안이 하나 이상으로 제시될 수 있는 비구조화, 혹은 반구조화 된 문제여야 한다(강정찬, 2015; 김재휘, 김동호, 2016; 송해덕, 2007; 임철일 외, 2014; Froyd, 2008; Shen et al., 2015). 본 연구에서는 이와 같은 교과-소프트웨어융합 수업의 문제 설계 단계에서 고려해야 할 사항들을 수업설계모형과 원리를 통해 체계적으로 설명하고 구체적인 예시를 제공하고 있으므로, 교수자가 수업의 주제와 융합 문제를 선정하고 설계하는 데 유용하게 활용할 수 있을 것이라 판단된다.

셋째, 각 수업 활동 단계에 따른 학습 지원이 적절히 제공되어야 한다. 개념모형의 구성요소인 ‘교과-소프트웨어융합 수업 활동’은 문제 이해, 아이디어 생성, 논리화(추상화), 문제해결(자동화)의 하위 요소를 포함하고 있으며, 이러한 활동은 ‘교과-소프트웨어융합 학습지원’ 구성요소를 통해 지원받을 수 있다. 따라서 적절한 학습 지원을 위해 교수자는 수업 활동에 대해 충분한 사전 안내를 제공하고, 학습 지원을 위해 협력 활동이 이루어질 수 있도록 모듈을 구성하며, 학습자가 발산·수렴적 사고를 촉진할 수 있는 학습자료를 작성 및 배포하고, 수업 중 학습자가 유념해야 할 규칙을 제시할 필요성이 있다(김성일, 2012; 노지예, 이정민, 2018; 변현정, 2019; 손영수, 이광재, 2016; 임철일 외, 2014; 전용주, 2017; Firertien, 1982).

마지막으로 본 연구에서는 학습 활동 내용에 초점을 맞춘 일반적인 수업모형이 아닌 수업의 설계를 위해 고려해야 하는 구체적이고 핵심적인 내용과 절차를 안내하고 있는 수업설계모형을 제공하고 있다. 앞서 소프트웨어를 활용한 융합 교육의 수업 모형으로서 CT-CPS모형 및 CT-STEAM모형 등에 대한 연구가 이루어졌으나, 이들은 한 모듈의 수업 내에서 이루어지는 교수자와 학습자의 학습 활동 내용에 초점을 맞춘 ‘수업 모형’으로서, 교수자의 입장에서 수업의 설계를 위해 고려해야 하는 내용과 절차를 안내하는데 한계를 가지고 있었다. 이에 본 연구에서는 실제 수업 설계를 수행하는 교수자의 입장에서 분석 단계부터 설계 및 개발, 평가까지의 거시적인 교과-소프트웨어융합 수업 설계를 위한

수업설계 모형과 핵심 원리를 제시하였고, 더불어 미시적인 층위에서 수업 설계 활동과 상세지침을 포함하였다. 또한 도출된 개념 모형, 절차 모형, 수업설계 원리를 토대로 학교 현장의 교사가 교과-소프트웨어융합 수업을 설계하고 실제 수업을 실시하였으며, 결과적으로 확인한 학습자와 교수자의 유의미한 반응을 통해 모형과 원리의 가능성을 확인할 수 있었다.

#### 나. 교과-소프트웨어융합 수업의 지원

본 연구에서 개발한 모형의 효과를 확인하기 위해 모형을 바탕으로 수업 설계를 하고 실제 수업 현장에 적용시켜 교수자의 반응과 학습자의 반응을 확인하였다. 교수자는 수업설계모형과 원리가 수업설계를 하는데 있어 매우 도움이 되었으며, 수업 설계를 위해 어떤 것을 고려해야 하는지를 구체적으로 제시하고 있어 도움이 되었다는 의견을 주었다. 절차모형에 대해서는 개념모형의 다양한 요소를 어떠한 순서대로 고려하여 수업을 설계해야 하는지 단계별로 자세히 제시하고 있어 수업 설계에 도움이 되었음을 밝혔다. 모형을 적용하여 진행된 수업에 대한 학습자 반응을 분석한 결과 창의적 문제해결력이 유의한 수준( $p < .05$ )에서 차이를 보였으며, 학습자 면담을 통해 교과-소프트웨어융합 수업 활동이 학습에 도움이 되었다는 학생들의 의견을 확인할 수 있었다. 이러한 교과-소프트웨어융합 수업을 실행하고 이를 효과적으로 지원하기 위한 수업 전략에 대해 논의해 보면 다음과 같다.

첫째, 수업을 위한 인프라 및 학교 내외의 환경 요인 등 제반 여건이 마련되어야 한다. 교과-소프트웨어융합 수업을 원활히 실시하기 위해서는 모듈별로 최소 1대 이상의 PC 혹은 태블릿 PC가 요구되며, 프로그래밍 소프트웨어 및 학습 플랫폼에 접근이 가능하도록 유무선 인터넷 장비 및 인프라가 필수적이다. 또한 코로나감염병-19로 인한 사회적 거리두기로 인해 온라인 비대면 수업의 실시가 필요한 경우, 학습자들은 집에서 사용할 수 있는 개인별 PC와 인터넷 연결 환경이 필요하다. 본 연구에

서는 코로나감염병-19로 인한 사회적 거리두기로 인해 불가피하게 온라인 수업으로 진행하게 되었으나, 수업 사전 연구자와 교수자가 논의하여 학습자들의 가정용 PC 구비 현황 및 인터넷 연결 여부, 소프트웨어 설치 여부를 파악하고 수업 진행 내용을 사전에 공지하여 장비 및 제반 여건이 부족하지 않도록 보완하는 과정을 거쳐 원활한 수업을 진행할 수 있었다. 이러한 환경 요인과 학습 도구는 교사의 교과-소프트웨어융합 수업 설계 및 실행에 큰 영향을 미친다. 예컨대 학교에 네트워크 인프라가 갖추어지지 않았거나, 가용할 수 있는 PC가 부족하다면 교사는 수업 설계에 앞서 학습 자원과 도구를 확보하는 데 더 큰 시간과 노력을 들여야 할 것이다. 또한 온라인으로 수업을 진행할 경우, 학습자들의 가정에 개인 PC가 구비되어 있고 화상회의 및 프로그래밍 소프트웨어가 설치되어 있는지, 와이파이 등 인터넷 연결이 가능한지를 확인해야 한다. 따라서 교수자는 교과-소프트웨어융합 수업 설계를 위해 학교 및 교실, 혹은 가정에 갖추어져 있는 학습 자원 및 도구를 확인하고 이에 맞추어 수업 내용 및 활동을 설계해야 한다.

둘째, 교과-소프트웨어융합 수업을 진행하기 이전에 소프트웨어 프로그래밍에 대한 사전 지식이나 경험을 쌓을 수 있도록 선행 수업을 실시할 필요성이 있다. 사전 지식은 새로운 기능이나 정보를 배우기 위해 요구되는 사전에 필요한 지식을 의미하며(허인숙, 2002), 교과-소프트웨어융합 수업을 원활히 진행하기 위해서는 학습자들이 소프트웨어에 대한 기본적인 사전지식을 쌓을 수 있는 기회를 제공해야 한다. 사전 학습이 된 학습자는 소프트웨어에 대한 지식을 바탕으로 교과-소프트웨어융합 문제를 분석하고 해결하는 활동에 보다 집중할 수 있으며, 학습자 중심의 능동적인 수업을 이끌어 낼 수 있다(Bergmann & Sams, 2012). 한편, 사전 학습이 부족한 학생들은 바로 본 프로그램을 시작하는 데 어려움을 겪으며, 프로그래밍 과정에서 높은 수준의 인지부하를 경험하고 전체적으로 학습에 어려움을 느낄 수 있다(신윤희, 정효정, 송종숙, 2019; 이재호, 장준형, 2017). 또한 소프트웨어 프로그래밍에 대한 기초적인 사전 지식이나 경험이 부족한 학생이 많은 경우, 이 학생들로 인해 수업 과정

이 지체되고 소프트웨어 교육이 제대로 이루어지지 않는 결과를 초래할 수 있다(이재호, 장준형, 2017). 이러한 측면을 고려하였을 때, 효과적인 교과-소프트웨어융합 수업을 위해서는 학습자들이 사전에 기본적인 소프트웨어의 기능과 정보에 대한 지식 및 경험을 갖추 수 있도록 충분한 사전 학습을 실시하여 출발점 지식을 갖추도록 기회를 제공할 필요성이 있다.

셋째, 수업 설계 시 실생활의 문제 상황을 분석하고 아이디어를 구조화 시킨 후 이를 논리적으로 개념화 시켜 프로그래밍 소프트웨어를 통해 구현시키는 일련의 과정을 효과적으로 실시하고 학습자의 부담감을 줄이기 위해서 사용하기 쉽고 다양한 소프트웨어 도구 및 수렴·발산적 사고 지원 도구를 활용하도록 계획해야 한다. 본 수업설계 모형에서 소프트웨어 및 사고지원 도구는 수업 내용을 전달하기 위해 학습자가 활용하는 교수 매체로서, 교수자는 학습자의 경험, 관심, 능력 수준, 학습 양식 등을 고려하여 선정해야 한다(전형기, 김영식, 2018). 본 연구에서 개발한 수업설계모형과 원리를 적용한 실제 수업에서는 문제 이해, 아이디어 생성, 논리화, 문제해결의 교과-소프트웨어융합 수업 활동의 목표와 단계별 내용 및 초등학교 고학년인 학습자들의 수준을 고려하여 소프트웨어 및 사고 지원도구를 선정하였다. 문제 이해 및 아이디어 생성 단계에서는 수업 이전의 사회과 수업과 연계된 프로젝트의 내용과 수업에서 제시된 문제 상황과 관련한 자료를 수집하여 문제를 정의하고, 모둠원들과의 아이디어 생성을 위해 패들렛(Padlet)을 활용하였다. 이때 브레인스토밍과 ALU기법과 같은 사고 지원도구를 활용하여 아이디어를 발산 및 수렴적 사고를 지원하였다. 논리화 단계에서는 모둠원 끼리 의견을 교환하며 아이디어를 구조화시키고 스토리보드를 작성할 수 있도록 협업문서작성 도구인 구글 독스(Google docs)를 선정하고 양식을 제공하였다. 마지막으로 문제해결 단계에서는 스토리보드와 알고리즘을 구현시키기 위해 초등학생도 손쉽게 이해하고 사용할 수 있는 블록형 프로그래밍 언어인 엔트리(Entry)를 활용하였다. 이러한 도구 선정과 활용에 대한 수업 후 학습자 심층 면담 결과 학습자들은 소프트웨어 도구 및 사고 지원도구

활용에 큰 어려움이 없었으며, 도구 활용이 수업에 대한 동기와 재미를 더해주었다고 응답하였다. 이는 교수 매체 선정이 교수 목표의 유형, 학습 내용의 특성, 교수학습방법 등을 고려하여 교수매체를 선정해야 한다고 주장한 리저와 가네의 주장과도 유사한 맥락이라 할 수 있다(박성익, 임철일, 이재경, 최정임, 2007).

## 다. 연구의 이론적, 실천적 함의

본 연구를 통해 다음과 같은 이론적 실천적 함의를 찾아볼 수 있다.

첫째, 개념모형, 절차모형 및 수업설계 원리의 개발을 통해 교사가 창의적 문제해결을 위한 컴퓨팅 사고력 기반 교과-소프트웨어융합 수업을 종합적이고 체계적으로 설계할 수 있게 하였다. 이는 컴퓨팅 사고력을 중심으로 수업의 효과성을 측정하거나(구덕희, 우석준, 2018; 김경규, 이종연, 2016; 이은경, 2012) 수업 모형 (배영권, 2007; 박주성, 김종한, 김태영, 2011; 전용주, 2017; 함성진, 김순화, 송기상, 2014) 및 교육 프로그램(노지예, 이정민, 2018; 류미영, 한선관, 2015; 이영준, 임웅, 이은경, 2010)만 제시하는 연구와 다르게 교사가 수업을 설계하기 위해 고려해야 할 전체적인 수업설계 모형을 제시하고 핵심 원리와 상세 지침을 개발하였기 때문이다. 특히 최근의 소프트웨어 교육 모형들은 교수설계를 위한 모형이 아니라 교수자가 수업상황에서 어떤 절차와 과정을 거쳐 수업을 진행하는지를 알려주는 수업모형의 형태를 띠고 있기 때문에, 컴퓨터교육 전공자가 아닌 일반 교사들이 보편적으로 활용하기에는 어려움이 있을 가능성이 높다(임정훈, 2018). 이러한 점을 고려하여 본 연구에서는 학교 현장에서 수업을 이끌어 나가는 교사가 학습자, 학습내용, 수업 환경, 수업 맥락 등의 핵심적인 수업 구성 요소들의 분석에서부터 교육과정을 재구성하고 교과-소프트웨어융합 문제와 활동 및 학습지원 전략을 지원하는 총체적인 수업 설계 과정을 안내하는 수업 설계 개념모형과 절차모형을 개발하였으며, 이를 세부적으로 설명하는 수업설계 원리와 지침을 도출하였다.

둘째, 교과-소프트웨어융합 수업 관련 연구에 대한 이론적 고찰을 바탕으로 진행된 설계·개발 연구의 사례를 제공해 줄 수 있다. 최근까지의 연구는 소프트웨어 교육이나 컴퓨팅 사고력, 프로그래밍 교육 관련 변인들 간 관계나 효과를 설문조사 하는 형식으로 검증하는 조사 연구와 교육 프로그램을 개발하여 효과성을 알아보는 개발 및 실험 연구가 주류를 이루고 있는 반면(이애화, 2018), 소프트웨어 교육을 위한 교수학습활동이 체계적으로 이루어질 수 있도록 수업 설계의 이론에 대해 심도 있게 고찰하고(임정훈, 2018), 이를 바탕으로 교수학습모형을 개발하여 그 효과성을 탐구하는 형식의 설계·개발연구는 부족한 상황이다. 본 연구는 소프트웨어와 컴퓨팅 사고력, 융·복합적 사고, 창의성과 창의적 문제해결력과 관련한 이론과 연구, 관련 문헌을 탐구 및 분석하여 수업설계 모형과 원리를 도출한 후, 내적·외적 타당화를 통해 교과-소프트웨어융합 수업의 준비와 실행을 위한 핵심 요소와 실제 고려해야 할 내용들을 정리하였다. 도출한 연구 결과는 교수자가 수업 설계에 대한 전체적이고 종합적인 시각을 가질 수 있도록 안내를 제공해준다는 점에서 의의를 갖는다.

셋째, 본 연구는 소프트웨어 교육이 기존의 실과나 정보 교과에만 한정하는 것이 아닌, 국어, 사회, 수학, 과학 등의 다양한 교과와 융합하여 활용될 수 있음에 대한 가능성과 아이디어를 제공해 줄 수 있다. 타 교과와의 융합적 맥락 없이 단순히 프로그래밍에 대한 내용을 학습하는 것은 그 효과가 대부분 컴퓨팅 사고력이나 프로그래밍 능력의 향상에만 한정되어 있는 경우가 많다(구덕희, 우석준, 2018; 김성열, 2020; 이은경, 2012; 허경, 2019). 따라서 소프트웨어 교육은 과학, 수학 등 교과적 지식에 근거한 실생활 문제 상황에서 프로그래밍 기술을 활용할 수 있도록 제시되어야 한다(윤진아, 남윤경, 조운석, 2020). 본 연구의 교과-소프트웨어융합 수업설계 모형은 국어와 사회, 그리고 소프트웨어 단원을 포함하고 있는 실과의 교과 융합을 통해 수업 전반에 걸쳐 융합적, 창의적 문제해결 역량을 기를 수 있도록 수업 설계의 방향을 안내하기 위해 개발되었으며, 교수자와 학습자의 긍정적 반응을 통해 다양한 교과 지식과

소프트웨어의 융합 수업 활동의 효과성을 검토하였다. 특히 수업 설계 시 참고할 수 있는 과목별, 활동별 예시와 고려해야 하는 요소들을 제공함으로써 교과-소프트웨어융합 수업의 확대 및 응용 가능성에 기여할 수 있다는 점에서도 의의를 찾아볼 수 있다.

## 2. 결론 및 제언

### 가. 결론

본 연구는 교과-소프트웨어융합 수업을 설계하기 위한 개념 모형, 절차 모형, 수업설계 원리를 개발하고 실제 학교 현장의 적용을 통한 교수자와 학습자의 반응과 효과를 확인하였다. 연구 결과를 바탕으로 다음과 같은 결론을 내릴 수 있다.

첫째, 교과-소프트웨어융합 수업 설계는 교과-소프트웨어융합 수업 활동을 중심으로 이루어지며, 교육과정 재구성 및 교과-소프트웨어융합 문제 설계가 전반적인 활동의 방향성을 설정하고, 산출물에 대한 공유 및 평가로 마무리 되는 구조를 가진다. 이와 같은 교과-소프트웨어융합 수업 설계는 선형적인 형태 보다는 개발 및 실행 과정에서 지속적이고 반복적인 수정과 개선이 이루어질 수 있는 유기적인 형태를 가지며, 교수자의 상황에 따라 필요시 일부 과정을 축소하거나 확대할 수 있다. 교과-소프트웨어융합 수업 활동을 중심으로 설계된 수업 설계 과정은 프로그래밍 소프트웨어 자원 및 하드웨어 자원, 이러닝 플랫폼, 수업 발산적 사고를 지원할 수 있는 사고 지원도구 및 소프트웨어의 지원을 전제로 전개된다. 이때 활용할 수 있는 자원과 도구의 상황을 고려하여 수업 설계에 반영할 필요가 있다. 예컨대 개인별 PC나 태블릿 PC의 수가 한정되어 있다면, 모듈별로 한 대씩 배정하여 모듈원이 돌아가며 프로그래밍을 수행하도록 안내할 수 있다. 또한 교과-소프트웨어수업 활동 설계 과정은 학교 및 동료교사의 지원과 협조, 교과-소프트웨어융합 수업에

대한 분위기, 기술적 인프라 등의 포괄적인 학습 환경 요인을 바탕으로 이루어진다.

둘째, 본 연구를 통해 개발된 수업 설계 개념모형, 절차모형 및 설계 원리를 바탕으로 실제 수업 현장에 적용하여 교과-소프트웨어융합 수업을 실시하였으며, 이에 대한 교수자와 학습자의 긍정적인 반응을 확인할 수 있었다. 교과-소프트웨어융합 수업설계 원리 중 ‘교과-소프트웨어융합 교육과정’ 구성요소에 포함된 원리와 세부지침을 반영하여 융합할 교과 및 단원의 학습 내용을 바탕으로 교육과정을 하나의 맥락으로 재구성하여 설계한 수업에 대한 학습자들의 반응을 심층 면담을 통해 확인한 결과, 학습자들은 다양한 교과목의 내용지식과 소프트웨어를 융합한 교과-소프트웨어융합 수업에 대해 긍정적인 반응을 보였으며, 향후 다른 교과목과의 소프트웨어융합 수업을 학습할 수 있다면 도움이 될 것 같다는 응답을 하였다. 수업설계 원리 중 ‘교과-소프트웨어융합 수업활동’ 구성요소에 포함된 문제이해, 아이디어 생성, 논리화, 문제해결 활동을 지원하기 위한 원리와 지침을 토대로 수업을 설계하고, ‘교과-소프트웨어융합 학습지원’에 포함된 모듈구성 및 자료 지원 관련 원리를 반영하여 수업을 실시한 결과, 학습자들의 창의적 문제해결력이 유의미하게 상승한 것을 확인하였으며, 전반적인 문제해결 활동이 흥미로웠고, 학습에 도움이 되었다는 응답을 확인할 수 있었다. 또한 ‘공유 및 평가’ 구성요소를 반영하여 수업 중 산출물의 공유와 평가 활동을 실시하였고, 학습자들은 서로의 산출물을 평가하고 다른 학습자들의 피드백을 받을 수 있어서 유익했다는 의견을 주었으며, 이를 산출물에 반영하여 수정할 수 있어 완성도를 높일 수 있어 도움이 되었다는 긍정적인 반응을 보였다. ‘교과-소프트웨어융합 학습도구’의 원리를 반영하여 수업 활동에 따라 아이디어를 생성하고 수렴하는 활동을 지원하는 다양한 소프트웨어를 활용하였고, 학습자들은 이러한 학습 자원 및 도구를 활용을 통해 아이디어를 구성하고 문제 해결 과정의 큰 줄기와 세부적인 활동을 정리할 수 있어서 재미있고 도움이 되었다는 긍정적인 반응을 보였다. 이러한 교과-소프트웨어융합 수업의 수업설계 원리와 개념모형, 절차모형에 대해 교수

자의 반응을 확인한 결과 전체적인 수업 설계를 위해 고려해야 할 구성 요소와 구성요소들의 관계를 개념적으로 알 수 있었고, 수업 설계의 과정과 단계를 구체적, 세부적으로 알 수 있어서 도움이 되었다는 의견을 확인할 수 있었다.

셋째, 본 연구를 통해 개발된 수업설계 모형과 원리를 실제 수업 현장에 적용시켜 본 결과, 학습자 수준차이를 고려하여 다양하고 구체적인 대안과 사례를 추가적으로 제공할 필요성이 있음을 확인할 수 있었다. 실제 학교 현장에서 수업설계 모형과 원리를 바탕으로 수업을 진행하였을 때, 교과-소프트웨어융합 수업 활동을 위해 다양한 수준의 학습자로 모둠을 구성한 결과 프로그래밍을 잘 하고 경험이 많은 학습자가 대부분의 프로그래밍 작업을 수행 하는 것으로 나타났다. 학습자 면담을 통해서도 학습자 수준차이가 큰 모둠의 학생들이 프로그래밍을 잘 하는 학습자가 전적으로 작업을 주도할 수밖에 없어 불합리하게 느껴졌고, 활동을 진행하는 것이 다소 어려웠다는 의견을 확인할 수 있었다. 이러한 결과를 반영하여 학습자들 모두가 효과적으로 교과-소프트웨어융합 수업 활동에 참여하기 위해서 협력활동을 위한 모둠 구성과 관련한 일부 원리와 모형의 개선과 추가적인 노력이 필요할 것으로 보인다.

## 나. 연구의 제한점 및 추후 연구를 위한 제언

본 연구의 제한점과 추후 연구를 위한 제언은 다음과 같다. 첫째, 외적 타당화를 위한 실제 수업 진행 시 코로나감염병-19의 확산으로 인하여 불가피하게 비대면 수업 형식으로 적용되었다. 교과-소프트웨어융합 수업 특성상 패들렛, 구글 독스 등의 온라인 협업 도구 활용이 가능하고 EPL 플랫폼에 산출물을 탑재하여 댓글 형식으로 공유 및 평가가 이루어져 대부분의 수업설계 원리와 모형을 적용할 수 있었으나, 동료 프로그래밍 활동의 경우 온라인상에서의 적용에 어려움이 있었다. 오프라인 수업의 경우 한 모둠에서 서로 자리를 바꾸어가며 운전자와 네비게이터의 역할을 할 수 있지만, 본 연구에서는 온라인 수업을 통해 운전자의 화면

을 온라인 화상회의 프로그램의 화면 공유 기능을 활용하여 진행되었기 때문에, 활동 중간 역할을 바꾸는 것에 어려움이 있었다. 따라서 향후에는 온라인상에서의 협력적 프로그래밍 활동이 원활히 진행될 수 있도록 동시에 프로그래밍을 할 수 있는 플랫폼을 탐색하고 선정하여 추가적인 연구를 진행할 필요가 있다.

둘째, 수업 실행 전 선수 학습의 기회가 충분히 제공되지 못하였다. 본격적인 교과-소프트웨어융합 수업이 적용되기 이전 학습자들에게 기본적인 프로그래밍 소프트웨어에 대한 수업을 제공하였으나, 학습자 면담 결과 충분한 선수 학습 기회가 좀 더 주어졌다면 보다 높은 수준의 산출물을 개발할 수 있었을 것이라는 의견을 확인할 수 있었다. 이러한 측면을 고려하였을 때, 효과적인 교과-소프트웨어융합 수업을 위해서는 학습자들이 사전에 충분한 선수 학습 기회를 제공받을 수 있도록 정규 교과 시간에 소프트웨어 프로그래밍에 대한 내용을 편성하고, 정규 교과 이외에도 방과 후 학교에서 소프트웨어 교육을 실시하거나 소프트웨어 대회를 개최하는 등 다양한 소프트웨어 관련 경험을 제공하여 학습자들의 소프트웨어 활용 수준을 향상시키는 방법을 강구해 볼 필요성이 있다.

셋째, 학습자들의 프로그래밍 수준이 상이하야 수업 활동에 대한 참여율의 차이를 보였다. 현장 적용 시 학습자 수준차이를 고려하여 다양한 수준의 학생들을 하나의 모둠으로 구성하여 수업을 설계하였으나, 이러한 구성에 어려움을 느낀 학생들이 많았다. 학습 활동에 대한 학습자 면담 결과, 학습자들은 모둠 내에서 프로그래밍을 잘 하는 학생들이 대부분의 활동을 주도하였고, 이로 인해 소프트웨어를 직접 운용해 볼 수 있는 기회가 모두에게 제공되지는 못하였으며, 프로그래밍을 담당한 학생도 부담이 컸다는 의견을 확인할 수 있었다. 따라서 교과-소프트웨어융합 수업 시 학습자 수준 차이를 고려한 모둠 구성에 대한 새로운 방법을 제시하고 이에 대한 연구를 진행해 볼 필요성이 있다.

이러한 내용을 바탕으로 추후 연구를 위한 제언을 제시하면 다음과 같다. 첫째, 교과-소프트웨어융합 수업설계 모형을 바탕으로 설계한 수

업이 영향을 미치는 요인을 다양하게 확인할 필요가 있다. 본 연구에서는 교과-소프트웨어융합 수업설계 모형의 효과에 대해 창의적 문제해결력과 학업적 자기효능감의 향상 여부를 확인하였으나, 향후 연구에서는 학습 태도, 학습 동기 및 협력적 자기효능감 등의 추가적인 요인을 고려하여 보다 다양한 측면에서 교과-소프트웨어융합 수업의 효과성을 확인할 필요가 있다. 둘째, 본 모형을 적용한 수업에서는 수업 후 성찰일지 작성을 통해 수업 내용을 성찰할 수 있는 기회를 제공하였으나, 단계별 산출물이 나오므로 이를 누적하여 기록이 가능한 교과-소프트웨어융합 수업의 특성을 고려하여 오픈 포트폴리오 등 다양한 성찰의 방법을 제공하고 이에 대한 추가적인 연구의 노력이 필요하다. 셋째, 본 모형을 적용한 수업이 진행된 학교는 다른 일반 학교들에 비해 상대적으로 활용할 수 있는 소프트웨어 및 하드웨어 자원이 잘 구축되어 있는 환경이었으며, 비대면 수업 시에도 수업에 참여한 모든 학생들이 개인용 PC를 구비하여 참여가 가능하였다. 본 모형을 바탕으로 설계한 교과-소프트웨어융합 수업이 학교 현장에서 원활히 수행되기 위해서는 가용 자원이 부족한 여건에서도 활용될 수 있어야 한다. 따라서 인프라가 부족한 환경에서의 효과적인 수업 운영을 위한 교수·학습 전략 및 효과적인 수업 환경 구축, 학교 및 기관의 협력을 통한 시스템 마련, 나아가 교과-소프트웨어융합 수업을 위한 교사 연수 및 예비교사 소프트웨어 역량 강화를 위한 프로그램 개발 등에 대한 추가적인 연구가 필요하다.

## 참 고 문 헌

- 강갑원 (2015). 융합인재교육의 원류, 변천 및 그 정체성의 탐색. **영재와 영재교육**, 14(2), 5-29.
- 강정찬 (2015). 창의·융합 교육을 위한 수업설계원리 개발. **교육방법연구**, 27(3), 276-305.
- 강호감, 김태훈 (2014). 초등과학영재의 창의적 문제해결력 향상을 위한 융합인재교육 (STEAM) 프로그램 개발. **영재교육연구**, 24(6), 981-994.
- 교육과학기술부 (2011). **융합인재교육(STEAM) 활성화 방안**. 융합인재교육(STEAM) 수도권 설명회(2011. 7. 11.). 한국창의과학재단.
- 교육부a. (2015). **소프트웨어 교육 운영 지침**. 서울: 교육부.
- 교육부b. (2015). **초·중등학교 교육과정 총론**. 서울: 교육부
- 구덕희, 우석준 (2018). 마이크로비트 기반의 창의 컴퓨팅 교육 프로그램 개발. **정보교육학회논문지**, 22(2), 231-238.
- 구양미, 김영수, 노선숙, 조성민 (2006). 창의적 문제해결을 위한 웹기반 교수-학습 모형과 학습 환경 설계: 수학교과에서의 예시를 중심으로. **교과교육학연구**, 10, 209-234.
- 구재훈, 김태영 (2018). 융합형 CT-FL 수업 모형 기반 초등 소프트웨어 융합 교육 지도의 실제. **한국컴퓨터교육학회 학술발표대회논문집**, 23(1), 171-174.
- 구재훈, 전용주, 김태영 (2016). 초등 정보영재의 창의적 문제해결력 향상을 위한 CT-CPS 프레임워크 기반 수업 콘텐츠의 개발 및 적용. **한국초등교육**, 27(2), 339-357.
- 권정인 (2013). **Computational Thinking 기반의 교수-학습이 학습자의 창의적 문제해결에 미치는 효과성 연구**. 미간행 박사학위논문, 성균관대학교 대학원.
- 김경규, 이종연 (2016). 컴퓨팅 사고력 기반 프로그래밍 학습의 효과성 분석. **컴퓨터교육학회 논문지**, 19(1), 27-39.
- 김동만, 이태욱 (2019). 컴퓨팅 사고력에서 추상화의 고찰. **한국컴퓨터정보학회 학술발표논문집**, 27(1), 309-312.
- 김명인 (2018). **소프트웨어 융합 수학수업이 초, 중학생의 수학적 태도, 수학적 창의성 및 수학적 문제해결력에 미치는 영향**. 미간행

- 석사학위논문, 아주대학교 교육대학원.
- 김민영 (2006). **과학 영재의 창의적 문제해결력 향상을 위한 과학사 CPS 수업모형 개발과 유전영역 수업프로그램의 적용효과**. 미간행 박사학위논문, 한국교원대학교 대학원.
- 김민정, 이원규, 김자미 (2017). 컴퓨팅사고력 측정에 사용되고 있는 도구 분석을 통한 새로운 검사도구 개발방향 제시. **컴퓨터교육학회 논문지**, 20(6), 17-25.
- 김병조, 전용주, 김지현, 김태영 (2016). 초등정보융합영재의 창의성향상을 위한 Computational Thinking 기반 실생활 문제해결 수업콘텐츠 개발 및 적용. **교원교육**, 32(1), 159-186.
- 김선희 (2014). **디지털 매체를 활용한 포럼연극 수업설계 모형 개발**. 미간행 박사학위논문, 서울대학교 대학원.
- 김성기 (1993). 창의적 문제해결과 학습과제의 설계. **영재교육연구**, 2(1), 213-237.
- 김성열 (2020). 마이크로비트 (MicroBit) 를 활용한 IoT 코딩 교육 효과성에 대한 연구. **한국전자통신학회 논문지**, 15, 363-369.
- 김성욱 (2016). **모바일 탐구학습을 위한 수업설계 모형 개발 연구**. 미간행 박사학위논문, 서울대학교 대학원.
- 김성원, 정영란, 우애자, 이현주 (2012). 융합인재교육 (STEAM) 을 위한 이론적 모형의 제안. **한국과학교육학회지**, 32(2), 388-401.
- 김성일 (2012). 창의력 신장을 위한 설계 제품 제작 수업에서 예비교사들의 만족도에 대한 연구. **한국기술교육학회지**, 12(2), 139-159.
- 김성호, 김신웅, 강성미, 김용성 (2011). Treffinger 의 CPS 모형을 활용한 창의적 문제해결 방법. **한국정보과학회 학술발표논문집**, 38(2A), 59-62.
- 김수환 (2015). Computational Thinking 교육에서 나타난 컴퓨터 비전공 학습자들의 어려움 분석. **컴퓨터교육학회 논문지**, 18(3), 49-57.
- 김수환, 송상수, 임선명, 김지현 (2017). Computational Thinking 평가를 위한 엔트리 자동평가 시스템 개발. **한국컴퓨터교육학회 학술발표대회논문집**, 21(1), 53-56.
- 김순화, 함성진, 송기상 (2015). 컴퓨팅 사고력 기반 융합인재교육 프로그램의 효과성 분석 연구. **컴퓨터교육학회 논문지**, 18(3), 105-114.
- 김아영, 박인영 (2001). 학업적 자기효능감 척도 개발 및 타당화 연구. **교육학연구**, 39, 95-123.

- 김영채 (1999). **창의적 문제해결: 창의력의 이론, 개발과 수업**. 제5판. 서울: 교육과학사.
- 김영채 (2019). **창의력: 잠재능력의 이론과 교육**. 서울: 운성사.
- 김재휘, 김동호 (2016). 컴퓨팅 사고력 향상을 위한 초등 피지컬 컴퓨팅 교육과정 개발. **정보교육학회논문지**, 20(1), 69-82.
- 김정호 (2015). 융합교육 수업 사례 분석을 통한 핵심성과 융통성의 원리 탐색. **조형교육**, 56, 91-110.
- 김진수 (2012). **STEAM 교육론**. 서울: 양서원.
- 김진숙, 한선관, 김수환, 정순원, 양재명, 장의덕, . . . 전수진 (2015). **소프트웨어 교육 교수학습 모델 개발 연구** (수탁연구 CR 2015-35). 한국교육학술정보원.
- 김태형 (2015). 홀리즘에 근거한 STEAM 교육과정 재검토. **홀리스틱융합교육연구**, 19(4), 61-83.
- 김현철 (2015). **영국의 소프트웨어 교육 동향 및 한국교육에 주는 시사점** (연구보고 P2015-02-3). 한국교육개발원.
- 김현철, 김서영, 이현정, 김영일, 한건우 (2018). **소프트웨어 교육 현황과 개선 방향**. 소프트웨어 교육 현황과 개선 방향 간담회(2018.1.29.). 민주연구원.
- 김혜란, 최선영 (2019). 초등과학 수업에서 언플러그드 컴퓨팅을 활용한 소프트웨어 융합교육프로그램의 개발과 적용-우리 몸의 구조와 기능 단원을 중심으로. **생물교육**, 47(2), 213-222.
- 김혜란, 최선영(2019). 초등과학 수업에서 오조봇 코딩을 활용한 SW 융합교육프로그램의 개발과 적용. **초등과학교육**, 38(2), 234-243.
- 나일주, 정현미 (2001). 웹기반 가상교육 프로그램 설계를 위한 활동모형 개발. **교육공학연구**, 17(2), 27-52.
- 노상우, 안동순 (2012). 초등학교 융합인재교육 (STEAM) 의 발전 방향 모색. **교육종합연구**, 10(3), 75-96.
- 노지예, 이정민 (2018). 로봇 활용 소프트웨어 교육 프로그램 설계: 초등학생의 컴퓨팅사고력과 창의적 문제해결력을 중심으로. **교육공학연구**, 34(1), 1-37.
- 노희진, 백성혜 (2015). 스크래치를 활용한 고등학교 과학 수업에 대한 학생 인식. **한국과학교육학회지**, 35(1), 53-64.
- 류미영 (2015). **컴퓨팅 사고력(CT) 기반의 융합교육 프로그램 개발: 초등 수학교과를 중심으로**. 미간행 석사학위논문, 경인교육대학교

- 교육전문대학원,
- 류미영, 한선관 (2015). 초등 소프트웨어 교육을 위한 CT 교육 프로그램 개발. **정보교육학회논문지**, 19(1), 11-20.
- 류미영, 한선관 (2018). Computational Thinking 에 대한 정보교육 교사들의 인식 비교. **교육논총**, 38, 1-19.
- 목광수, 손정우, 배성문, 홍석영, 김용진 (2014). **코딩 교수법으로 진행하는 융·복합적 사고와의 만남**. 서울: 북스힐.
- 문대영 (2008). STEM 통합 접근의 사전 공학교육 프로그램 모형 개발. **공학교육연구**, 11(2), 90-101.
- 미래인재연구소 (2015). **창의컴퓨팅 이슈리포트**. Retrieved from <http://computing.or.kr/12067/미래인재연구소-이슈리포트-1호/>
- 박금주, 최영준 (2018). 비전공자를 위한 소프트웨어 교육 방향의 탐색. **교육문화연구**, 24(4), 273-292.
- 박만준 (2010). 지식과 융합과 경계적 사유. **철학연구**, 116, 111-132.
- 박성익, 임철일, 이재경, 최정임 (2007). **교육방법의 교육공학적 이해**. 서울: 교육과학사.
- 박영석, 구하라, 문중은, 안성호, 유병규, 이경윤, . . . 차윤경 (2013). STEAM 교사 연구회 개발 자료 분석: 융복합교육적 접근. **교육과정연구**, 31(1), 159-186.
- 박일수 (2013). 교육과정 재구성의 학습효과에 관한 메타분석. **교육과정연구**, 31(4), 141-164.
- 박주성, 김중환, 김태영 (2011). 창의적 문제해결력을 위한 가상 로봇 프로그래밍 수업 모형 연구. **한국컴퓨터교육학회 학술발표대회논문집**, 15(2), 301-305.
- 박태정 (2018). 창의적 문제해결 기반 퍼지컬컴퓨팅 교육프로그램 설계원리 개발 및 적용. **교육공학연구**, 34(3), 817-847.
- 박관우 (2016). 게임프로그래밍을 통한 초등 융합 소프트웨어교육 방안. **예술인문사회융합멀티미디어논문지**, 6(2), 27-34.
- 박현주, 김영민, 노석구, 이주연, 정진수, 최유현, . . . 백윤수 (2012). STEAM 교육의 구성 요소와 수업 설계를 위한 준거 틀의 개발. **학습자중심교과교육연구**, 12, 533-557.
- 배선아 (2011). 중학교 전기전자기술 영역의 활동 중심 STEM 교육프로그램 개발 및 적용. **대한공업교육학회지**, 36(1), 1-22.
- 배영권 (2006). **창의적 문제해결력 신장을 위한 유비쿼터스 환경의**

- 로봇프로그래밍 교육 모형.** 미간행 박사학위논문, 한국교원대학교 대학원.
- 배학진, 이은경, 이영준 (2009). 문제 중심 학습을 적용한 스크래치 프로그래밍 교수 학습 모형. **컴퓨터교육학회논문지**, 12(3), 11-22.
- 백윤수, 박현주, 김영민, 노석구, 이주연, 정진수, . . . 최종현 (2012). **융합인재교육 (STEAM) 실행방향 정립을 위한 기초연구** (연구보고 2012-12). 한국과학창의재단.
- 배윤주, 이정민 (2020). 초등 SW 융합교육의 효과에 대한 메타분석. **한국콘텐츠학회논문지**, 20(10), 247-258.
- 백연경 (2005). **스캠퍼 (SCAMPER) 기법을 활용한 발상지도가 디자인 수업에서 중학생들의 학습동기와 표현력에 미치는 효과.** 미간행 석사학위논문, 이화여자대학교 교육대학원.
- 변현정 (2019). 사회문제 개선을 위한 팀 기반 창의적 문제해결 학습경험 설계원리 개발. **창의력교육연구**, 19(1), 89-122.
- 서경혜 (2009). 교사들의 교육과정 재구성 실천 경험에 대한 사례연구. **교육과정연구**, 27(3), 159-189.
- 서영호, 염미령 (2016). 초등학교 소프트웨어 교육에서 동료 프로그래밍 교육 방법이 컴퓨팅 사고력과 창의성 신장에 미치는 효과 분석. **정보교육학회논문지**, 20(3), 219-234.
- 손영수, 이광재 (2016). IT 융합교육 활성화를 위한 Computational Thinking 수업 모형 설계. **한국전자통신학회 논문지**, 11(5), 511-522.
- 손원성 (2016). 블록기반 프로그래밍 코드의 수준 및 취약수준 측정방안. **정보교육학회논문지**, 20(3), 293-302.
- 송기상, 소효정 (2017). **ICE4ED 2017: 해외 소프트웨어 교육 동향 분석 및 시사점** (연구보고 RM2017-10). 한국교육학술정보원
- 송해덕 (2007). 창의적 문제해결력의 구성요인과 교수설계원리의 탐색. **열린교육연구**, 15(3), 55-73.
- 신윤희, 정효정, 송종숙 (2019). 소프트웨어 비전공 대학생을 위한 디자인 씽킹 기반 코딩교육에서의 학습경험 분석. **디지털콘텐츠학회지**, 20(4), 759-768.
- 신재한, 최은경 (2015). **STEAM 융합교육을 위한 교육 과정설계 및 적용.** 파주: 교육과학사.
- 신중호, 김민성, 최지은, 허유성, 이지은 (2015). **교육심리학.** 파주: 교육과학사.
- 안상진, 서영민, 이영준 (2012). 계산적 사고에 내재된 추상화의 특성 연구.

- 한국컴퓨터교육학회 학술발표대회논문집, 16(1), 43-46.
- 안성진 (2014). 소프트웨어교육 실현화 방안. Retrieved from [https://happyedu.moe.go.kr/happy/bbs/selectHappyArticleImg.do?bbsId=BBSMSTR\\_00000000192&nttId=3164](https://happyedu.moe.go.kr/happy/bbs/selectHappyArticleImg.do?bbsId=BBSMSTR_00000000192&nttId=3164)
- 연은경 (2013). 사례기반학습을 위한 사례 설계 원리 개발 연구. 미간행 박사학위논문, 서울대학교 대학원.
- 김경훈, 고상현, 김성식, 양재명, 이영준, 김현철, . . . 최숙영 (2015). 2015 개정 교과 교육과정 시안 개발 연구 I : 정보과 교육과정. 한국교육과정평가원.
- 우은경 (2019). 중학교 소프트웨어 교육 의무화에 따른 교육 방안. 미간행 석사학위논문, 한밭대학교 정보통신대학원.
- 유승희, 성용구 (2001). 초등교사를 위한 프로젝트 접근법. 서울: 양서원.
- 윤옥한 (2017). STEAM 교육을 위한 교수체제설계 모형 탐색: Design Thinking 을 중심으로. 교양교육연구, 11(1), 443-474.
- 윤진아, 남윤경, 조운석 (2020). 마이크로비트를 활용한 과학-소프트웨어 융합교육 프로그램 개발 및 적용. 컴퓨터교육학회 논문지, 23(6), 77-87.
- 이미화, 함성봉 (2018). 초등 융합 소프트웨어 교육을 위한 플립 러닝 기반 교수-학습 모형 개발 및 적용. 정보교육학회논문지, 22(2), 213-222.
- 이민희, 임해미 (2013). 수학사를 활용한 융합적 프로젝트기반학습 (STEAM PBL) 의 설계 및 효과 분석. 학교수학, 15(1), 159-177.
- 이소이, 노태천 (2011). STEM 교육을 위한 기술 수업설계 모형. 한국기술교육학회지, 11(3), 1-20.
- 이애화 (2018). 국내 소프트웨어교육 연구동향 분석. 교육정보미디어연구, 24(2), 277-301.
- 이영기, 손장호 (2016). '실과-STEAM-환경'프로그램이 초등학생의 창의적 문제해결력과 환경소양에 미치는 영향. 실과교육연구, 22(2), 15-31.
- 이영준, 백성혜, 신재홍, 유현창, 정인기, 안상진, . . . 전성균. (2014). 초중등 단계 Computational Thinking 도입을 위한 기초연구. 한국 과학 창의 재 단 . <https://scienceon.kisti.re.kr/srch/selectPORSrchReport.do?cn=TRKO201500007885>에서 검색
- 이영준, 임웅, 이은경 (2010). 창의적 문제해결력 향상을 위한 정보 교육 프로그램. 컴퓨터교육학회 논문지, 13(1), 1-8.

- 이영태, 남창우 (2015). 고등학교 창의, 융합 수업을 위한 교수, 학습모형 개발. **교육공학연구**, 31(3), 533-569.
- 이운발 (2005). **고등 사고력 함양을 위한 주제 스트랜드 중심의 사회과 통합교육과정 구성의 방략**. 미간행 박사학위논문, 경북대학교 대학원.
- 이은경 (2012). 계산적 사고 (Computational Thinking) 능력 향상을 위한 STEAM 교육 방안. **한국컴퓨터교육학회 학술발표대회논문집**, 16(1), 47-51.
- 이은아 (2018). **창의적 체험 활동에서의 코딩교육이 일반계 고등학생의 창의적 문제해결력과 자기효능감에 미치는 영향**. 미간행 석사학위논문, 한국교통대학교 교육대학원.
- 이재경 (2017). **특성화 고등학교 교과 융합 프로젝트 기반의 국어과 융합 수업 방안**. 미간행 석사학위논문, 한국교원대학교 교육대학원.
- 이재호, 장준형 (2017). 과학영재용 소프트웨어 코딩기반 메이커 교육 프로그램의 개발. **영재교육연구**, 27(3), 331-348.
- 이재호, 하희정 (2016). 소프트웨어 코딩기반 메이커교육용 교수-학습모형의 효과성 검증. **창의정보문화연구**, 2(2), 49-61.
- 이철현 (2015). 초등 소프트웨어 교육 방향 탐색 및 모델 개발. **한국실과교육학회지**, 28(4), 207-222.
- 이혜영, 이태욱 (2015). 컴퓨팅 사고력 신장을 위한 프로그래밍학습 기반 중학교 과학교과 프로그램 개발. **한국컴퓨터정보학회논문지**, 20(12), 181-188.
- 임유정 (2012). **수학 교과와 타 교과간의 융합 수업 모형 개발 및 적용 과정에 대한 연구**. 미간행 석사학위논문, 영남대학교 교육대학원.
- 임정훈 (2018). 소프트웨어 교육의 주요 이슈와 소프트웨어 교육 발전을 위한 교육공학자의 역할. **교육공학연구**, 34(3), 679-709.
- 임철일 (2012). **교수설계 이론과 모형**. 파주: 교육과학사.
- 임철일, 김성욱, 한형중, 서승일. (2014). 창의적문제해결을위한스마트지원시스템의수업적용: 미술대학 수업 사례, **아시아교육연구**, 15(3), 171-201
- 임철일, 연은경 (2006). 기업교육 프로그램 개발을 위한 사용자 중심의 래피드 프로토타입 방법론에 관한 연구. **기업교육과 인재연구**, 8, 27-50.
- 임철일, 윤순경, 박경선, 홍미영 (2009). 온라인 지원 시스템 기반의 “창의적 문제해결 모형” 을 활용하는 통합형 대학 수업 모형의 개발. **교육공학연구**, 25(1), 171-203.
- 임철일, 홍미영, 이선희. (2011). 공학교육에서의 창의성 증진을 위한 학습환경

- 설계모형. **공학교육연구**, 14(4), 3-10.
- 임화영, 윤혜경 (2019). 초등 교사의 과학과 교육과정 재구성 경험에 대한 사례 연구. **초등과학교육**, 38(1), 102-115.
- 전경원 (2006). **창의성 교육의 이론과 실제**. 서울: 창지사.
- 전성관, 서영민, 이영준 (2010). CPS 기반의 창의적 프로그래밍 수업 모형. **한국컴퓨터교육학회 학술발표대회논문집**, 14(2), 95-99.
- 전성관, 이영준. (2014). EPL 을 활용한 예술 중심의 STEAM 교육 프로그램. **한국컴퓨터정보학회논문지**, 19(4), 149-158.
- 전수진 (2017). 소프트웨어 교육에서의 컴퓨팅 사고력 신장을 위한 개발 중심 모형의 설계 및 효과. **정보교육학회논문지**, 21(6), 619-627.
- 전용주 (2017). **새로운 교육과정의 소프트웨어 교육을 위한 컴퓨팅 사고력 기반 창의적 문제해결(CT-CPS) 수업모형의 개발 및 적용**. 미간행 박사학위논문, 한국교원대학교대학원.
- 전윤식, 김정섭, 윤경미 (2003). 창의성 교육의 새로운 접근: 문제 찾기. **교육학연구**, 41(3), 215-238.
- 전형기, 김영식 (2018). 2015 개정 초등 교육과정의 소프트웨어 교육을 위한 피지컬 컴퓨팅 기반 교구 평가 준거 개발. **컴퓨터교육학회 논문지**, 21(5), 37-48.
- 정영식 (2017). **4차 산업혁명 시대의 소프트웨어 교육방안**. 한국정보화진흥원.
- 정예진 (2018). **컴퓨팅 사고력(CT)을 활용한 수학중심 STEAM 프로그램이 초등 수학 영재 학생의 융합적 사고력 및 협업 역량에 미치는 영향**. 미간행 석사학위논문, 고려대학교 교육대학원.
- 정정호 (2013). **디자인융합인재의 창의성에 대한 다중지능 및 사고양식 유형 연구**. 미간행 박사학위논문, 홍익대학교 대학원.
- 조연순, 성진숙, 이해주 (2008). **창의성 교육: 창의적 문제해결력 개발과 교육방법**. 서울: 이화여자대학교출판부.
- 조연순, 정혜영, 백은주, 최규리, 임현화 (2011). 교실수업 상황에서 학생 창의성을 촉진하는 교수· 학습 특성 탐색. **교육방법연구**, 23(3), 609-634.
- 조향숙, 김훈, 허준영. (2012). **현장 적용 사례를 통한 융합인재교육 (STEAM) 의 이해** (연구보고 OR2012-02-02). 한국교육개발원
- 주미경, 문중은, 송륜진 (2012). 수학교과와 융복합교육: 담론과 과제. **학교수학**, 14(1), 165-190.
- 주영주, 박수영 (2015). 소프트웨어 교육에서의 자기효능감, 교수매락,

- 학습접근방식과 학습만족도 간의 관계 규명. **학습자중심교과교육연구**, 15(8), 325-341
- 지경옥 (2017). 현장에서의 교육과정 재구성을 통한 실과교육. **한국실과교육학회 학술대회논문집**, 12, 131-136.
- 차현진, 김민하 (2020). 소프트웨어 교육의 창의적 문제해결 프로젝트에서 프로토타이핑 과제의 수준에 따른 효과 분석: 인지된 학업적 자기효능감과 문제해결력 중심으로. **학습자중심교과교육연구**, 20(3), 689-711.
- 최상덕 (2011). 21 세기 창의적 인재 양성을 위한 교육의 미래전략 연구 (연고보고 RR2011-01). 한국교육개발원.
- 최소영 (2017). 미래 공학자 만들기, 소프트웨어 초등교육으로부터. **공학교육동향**, 24(1), 22-25.
- 최숙영 (2016). 문제해결의 관점에서 컴퓨팅 사고력 증진을 위한 교수학습에 대한 연구. **컴퓨터교육학회 논문지**, 19(1), 53-62.
- 최정임 (2004). 사례 분석을 통한 PBL 의 문제설계 원리에 대한 연구. **교육공학연구**, 20(1), 37-61.
- 최희진 (2016). 일반계열 고등학생의 학업적 자기효능감 증진을 위한 집단상담 프로그램 개발. 미간행석사학위논문, 한국교원대학교 교육대학원.
- 태진미 (2010). 영재를 위한 문화예술 통합 교육의 필요성과 적용 방안. **순천향 인문과학논총**, 26, 241-273.
- 한국과학창의재단 (2012). **손에 잡히는 STEAM 교육: 무엇이 아이들을 즐겁게 하는가**. 서울: 한국과학창의재단.
- 한선관 (2015). **Creative thinking 이슈리포트**. Retrieved from <http://computing.or.kr/wp-content/uploads/2016/02/1.-%EC%9D%B4%EC%8A%88%EB%A6%AC%ED%8F%AC%ED%8A%B8-1%ED%98%B8.pdf>
- 함성진, 김순화, 박세영, 송기상 (2014). 융합적사고력 신장을 위한 초등학생용 CT 기반 융합인재교육 (CT-STEAM) 프로그램 개발. **컴퓨터교육학회 논문지**, 17(6), 81-91.
- 한영신 (2018). 문제중심학습 (PBL) 기반 프로그래밍 교육의 효과성: 컴퓨팅 사고력을 중심으로. **예술인문사회 융합 멀티미디어 논문지**, 8(7), 433-445.
- 허경 (2019). 비전공자 대상 Java 소프트웨어 교육 강좌에서 마이크로비트를

- 이용한 컴퓨팅적 사고과정 교육 방법. *실천공학교육논문지*, 11(2), 167-174.
- 허인숙 (2002). 사회과 교육에서 사전지식을 고려한 학습과 개념도의 활용. *시민교육연구*, 34(2), 235-254.
- 홍미영 (2012). *창의적 문제해결을 위한 이러닝 지원시스템 설계원리 개발연구*. 미간행 박사학위논문, 서울대학교 대학원.
- 홍정미 (2015). *스크래치를 활용한 소프트웨어융합교육 프로그램이 초등학생의 ICT소양과 소프트웨어인식에 미치는 영향*. 미간행 석사학위논문, 경인교육대학교 대학원.
- 홍현미 (2017). *융합인재교육 (STEAM) 을 위한 플립러닝 수업설계원리 개발 연구*. 미간행 박사학위논문, 서울대학교 대학원.
- 황지운, 황성온 (2017). 초등 소프트웨어 교육 연구 동향 분석: 국내 학술지 중심으로. *정보교육학회논문지*, 21(5), 509-525.
- 황지현 (2010). *창의적 문제해결력 신장을 위한 가상현실 학습공간의 설계*. 미간행 석사학위논문, 한국교원대학교 대학원.
- Amabile, T. M. (1983). The social psychology of creativity: A componential conceptualization. *Journal of personality and social psychology*, 45(2), 357.
- Bacon, D. R., Stewart, K. A., & Anderson, E. S. (2001). Methods of assigning players to teams: A review and novel approach. *Simulation & Gaming*, 32(1), 6-17.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads*, 2(1), 48-54.
- Basadur, M. S. (1994). Managing the creative process in organizations. In M. A. Runco (Ed.), *Problem finding, problem solving, and creativity* (pp. 237 - 268). Norwood, NJ: Ablex.
- Bergmann, J., & Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. International society for technology in education.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada.

- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, 4(2), 121-135.
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In *Design for teaching and learning in a networked world* (pp. 17-27): Springer.
- Capraro, R., Morgan, M., Morgan, James R, & SpringerLink. (2013). *STEM project-based learning [electronic resource] : An integrated science, technology, engineering, and mathematics (STEM) approach / edited by Robert M. Capraro, Mary Margaret Capraro and James R. Morgan*. (2nd ed., Other Books). Rotterdam: Sense.
- Chen, Y.-F., & Cheng, K.-w. (2009). Integrating computer-supported cooperative learning and creative problem solving into a single teaching strategy. *Social Behavior and Personality: an international journal*, 37(9), 1283-1296.
- Council, N. R. (2010). *Report of a workshop on the scope and nature of computational thinking*: National Academies Press.
- Cropley, A. J. (2001). *Creativity in education & learning: A guide for teachers and educators*: Psychology Press.
- Cropley, D. H. (2016). Measuring capacity for innovation in local government organizations. *Innovation*, 2, 31-45.
- Cross, J., Bartley, C., Hamner, E., & Nourbakhsh, I. (2013). *A visual robot-programming environment for multidisciplinary education*. Paper presented at the 2013 IEEE International Conference on Robotics and Automation.
- CSTA, I. (2011). Computational Thinking Leadership toolkit 1st edition. Retrieved from <http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>.
- Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing

- computational thinking in the classroom: a framework.
- Davis, L. L. (1992). Instrument review: Getting the most from a panel of experts. *Applied nursing research*, 5(4), 194-197.
- DeSchryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated approaches to teaching and learning. *Journal of Technology and Teacher Education*, 23(3), 411-431.
- Drake, S. M., & Burns, R. C. (2004). *Meeting standards through integrated curriculum*: ASCD.
- Ellis, T. J., & Levy, Y. (2010). *A guide for novice researchers: Design and development research methods*. Paper presented at the Proceedings of Informing Science & IT Education Conference (InSITE).
- Ernst, G. W., & Newell, A. (1969). *GPS: A case study in generality and problem solving*: Academic Press.
- Feldhusen, J. F., & Treffinger, D. J. (1983). *Creative thinking and problem solving in gifted education*. Dubuque, IA: Kendall-Hunt.
- Feldhusen, J. F. (1993). A conception of creative thinking and creativity training. *Nurturing and developing creativity: The emergence of a discipline*, 31-50.
- Firetten, R. L. (1982). *From basics to breakthroughs*. Buffalo, NY: DOK.
- Frederiksen, N. (1984). The real test bias: Influences of testing on teaching and learning. *American psychologist*, 39(3), 193.
- Froyd, J. E. (2008). White paper on promising practices in undergraduate STEM education. *Commissioned paper for the Evidence on Promising Practices in Undergraduate Science, Technology, Engineering, and Mathematics (STEM) Education Project, The National Academies Board on Science Education*.
- Gardner, H. (1993). Creating minds: An anatomy of creativity seen through the lives of Freud. *Einstein. Picasso. Stravinsky. Eliot. Graham, and Gandhi*. New York: Basic Books.
- Green, S. L., & Anid, N. M. (2013). *Training K-12 teachers in STEM education: A multi-disciplinary approach*. Paper presented at the 2013 IEEE Integrated STEM Education Conference (ISEC).
- Guilford, J. P. (1967). *The nature of human intelligence*. New York: NY:

McGraw-Hill.

- Guilford, J. P. (1975). Factors and factors of personality. *Psychological Bulletin*, 82(5), 802.
- Hambruch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin*, 41(1), 183-187.
- Hamza, M., & Griffith, K. G. (2006). *Fostering problem-solving & creative thinking in the classroom: cultivating a creative mind*. Paper presented at the National Forum of Applied Educational Research Journal-Electronic.
- Hutchins, N. M., Bissof, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., . . . Basu, S. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 29(1), 83-100.
- Harvard Graduate School of Education (2014). *Creative Computing*. Retrieved from <http://scratched.gse.harvard.edu/guide/curriculum.html>
- Isaksen, S. G., & Treffinger, D. J. (1985). Creative problem solving. *The Basic Course*. New York: Bearly Limited.
- Isaksen, S. G., & Treffinger, D. J. (2004). Celebrating 50 years of reflective practice: Versions of creative problem solving. *The Journal of Creative Behavior*, 38(2), 75-101.
- ISTE, CSTA, & NSF. (2011). Computational Thinking - Teacher Resources 2nd edition. Retrieved from [http://www.iste.org/docs/ct-documents/ct-teacher-resources\\_2ed-pdf.pdf?sfvrsn=2](http://www.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2)
- J., T. D., Isaksen, S. G., & Dorval, K. B. (1994). *Creative problem solving: and overview*. Norwood, NJ: Ablex.
- J., T. D., Isaksen, S. G., & Dorval, K. B. (2000). *Creative problem solving: an introduction* (3 ed.). Waco, TX: Prufrock Press.
- Jonassen, D. H. (1997). Instructional design models for well-structured and III-structured problem-solving learning outcomes. *Educational technology research and development*, 45(1), 65-94.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning

- during design and computing. *Computers & education*, 73, 26–39.
- Kong, S.-C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, 3(4), 377–394.
- Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 36–42.
- Krauss, J., & Prottzman, K. (2017). *Computational thinking and coding for every student: The teacher's getting-started guide*: Corwin Press.
- Lavonen, J., Autio, O., & Meisalo, V. (2004). Creative and Collaborative Problem Solving in Technology Education: A Case Study in Primary School Teacher Education. *Journal of technology studies*, 30(2), 107–115.
- Lin, Y.-T., Wang, M.-T., & Wu, C.-C. (2019). Design and implementation of interdisciplinary STEM instruction: Teaching programming by computational physics. *The Asia-Pacific Education Researcher*, 28(1), 77–91.
- Lynn, M. R. (1986). Determination and quantification of content validity. *Nursing research*.
- MacKinnon, D. W. (1978). *In search of human effectiveness*: Creative Education Foundation.
- Mayer, R. E. (1987). The elusive search for teachable aspects of problem solving. In *Historical foundations of educational psychology* (pp. 327–347): Springer.
- Mednick, S. (1962). The associative basis of the creative process. *Psychological review*, 69(3), 220.
- Minamino, K. i., & Kinoshita, T. (2010). A web-based system to support group idea generation for creative problem solving. In *Technological Developments in Education and Automation* (pp. 361–367): Springer.
- Mumford, M. D., Reiter-Palmon, R., & Redmond, M. R. (1994). Problem construction and cognition: Applying problem representations in ill-defined domains.
- Newell, A., Shaw, J. C., & Simon, H. A. (1962). *The processes of creative thinking*. Paper presented at the Contemporary Approaches to Creative Thinking, 1958, University of Colorado, CO, US; This paper

- was presented at the aforementioned symposium.
- Osborn, A. F. (1953). *Applied imagination*.
- Osborn, A. F. (1963). *Applied imagination: principles and procedures of creative thinking* (3 ed.). New York, NY: Charles Scribner's Sons.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. In: Basic Books, Inc.
- Parnes, S. J. (1967). *Creative behavior guidebook*: Scribner.
- Peterson, B. J. (2007). *An instructional design model for heuristics*. Citeseer.
- Puccio, G. J., Firestien, R. L., Coyle, C., & Masucci, C. (2006). A review of the effectiveness of CPS training: A focus on workplace issues. *Creativity and innovation management*, 15(1), 19–33.
- Puccio, G. J., Murdock, M. C., & Mance, M. (2005). Current developments in creative problem solving for organizations: A focus on thinking skills and styles. *Korean Journal of Thinking and Problem Solving*, 15(2), 43.
- Quigley, C. F., Herro, D., & Jamil, F. M. (2017). Developing a conceptual model of STEAM teaching practices. *School science and mathematics*, 117(1-2), 1-12.
- Reigeluth, C. M. (1983). Instructional design: What is it and why is it. *Instructional-design theories and models: An overview of their current status*, 1, 3-36.
- Reigeluth, C. M., & Carr-Chellman, A. A. (2009). *Instructional-design theories and models, volume III: Building a common knowledge base* (Vol. 3): Routledge.
- Reiser, R. A. (2012). What field did you say you were in? defining and naming our field. In R. A. Reiser & J. V. dempsey (Eds.), *Trends and issues in instructional design and technology* (3rd ed ed., pp. 1-7). Upper Saddle River, NJ: Merrill.
- Rennie, L., Venville, G., & Wallace, J. (2012). *Integrating science, technology, engineering, and mathematics: Issues, reflections, and ways forward*: Routledge.
- Repenning, A., Basawapatna, A., & Escherle, N. (2016). *Computational thinking tools*. Paper presented at the 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).

- Rhodes, M. (1961). An analysis of creativity. *The Phi Delta Kappan*, 42(7), 305-310.
- Richey, R. C., & Klein, J. D. (2007). Design and development research. *Mahwah, NJ*.
- Richey, R. C., & Klein, J. D. (2014). Design and development research. In *Handbook of research on educational communications and technology* (pp. 141-150): Springer.
- Richey, R. C., Klein, J. D., & Tracey, M. W. (2010). *The instructional design knowledge base: Theory, research, and practice*: Routledge.
- Robertson, J. (2011). The educational affordances of blogs for self-directed learning. *Computers & education*, 57(2), 1628-1644.
- Rubio, D. M., Berg-Weger, M., Tebb, S. S., Lee, E. S., & Rauch, S. (2003). Objectifying content validity: Conducting a content validity study in social work research. *Social work research*, 27(2), 94-104.
- Sanders, M. (2009). STEM, STEM education, STEMmania. the technology teacher. In: Virginia Tech Blacksburg.
- School, C. A. (2016). Computational Thinking. Retrieved from <https://community.computingschool.org.uk/files/8221/original.pdf>.  
<https://community.computingschool.org.uk/files/8221/original.pdf>
- Scott, G., Leritz, L. E., & Mumford, M. D. (2004). The effectiveness of creativity training: A quantitative review. *Creativity research journal*, 16(4), 361-388.
- Scott, T. (2016). The People's Code. Retrieved from <https://obamawhitehouse.archives.gov/blog/2016/08/08/peoples-code>
- Secretary, T. W. H. O. o. t. P. (2016). FACT SHEET: President Obama Announces Computer ScienceFor All Initiative. Retrieved from <https://www.whitehouse.gov/the-press-office/2016/01/30/fact-sheet-president-obama-announces-computer-science-all-initiative-0>
- Seechalio, T. (2017). Instructional strategies to support creativity and innovation in education. *Journal of Education and Learning*, 6(4), 201-208.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and*

- Information Technologies*, 18(2), 351-380.
- Seoane-Pardo, A. M. (2016). *Computational thinking beyond STEM: an introduction to "moral machines" and programming decision making in ethics classroom*. Paper presented at the Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality.
- Shen, J., Jiang, S., & Liu, O. L. (2015). Reconceptualizing a college science learning experience in the new digital era: a review of literature. *Emerging Technologies for STEAM Education*, 61-79.
- Simon, H. A. (1973). The structure of ill structured problems. *Artificial intelligence*, 4(3-4), 181-201.
- Smaldino, S. E., Lowther, D. L., Russell, J. D., & Mims, C. (2008). Instructional technology and media for learning.
- Smaldino, S. E., Lowther, D. L., Russell, J. D., & Mims, C. (2008). *Instructional technology and media for learning (8th ed.)*. Upper Saddle River, NJ: Pearson Education.
- Stohlmann, M., Moore, T. J., & Roehrig, G. H. (2012). Considerations for teaching integrated STEM education. *Journal of Pre-College Engineering Education Research (J-PEER)*, 2(1), 4.
- Taljaard, J. (2016). A Review of Multi-Sensory Technologies in a Science, Technology, Engineering, Arts and Mathematics (STEAM) Classroom. *Journal of learning Design*, 9(2), 46-55.
- Taylor, C. W. (1988). Various approaches to and definitions of creativity. In *The nature of creativity: Contemporary psychological perspectives* (R. J. Sternberg ed., pp. 99 - 124). Cambridge, UK: Cambridge University Press.
- Torrance, E. P. (1972). Can We Teach Children to Think Creatively? *Journal of Creative Behavior*, 6(2), 114-143.
- Torrance, E. P. (1974). *the torrance tests of creative thinking-norms-technical manual research edition-verbal tests, forms A and B-figural test, forms A and B*. Princeton, NJ: Personnel Press.
- Treffinger, D. J., & Isaksen, S. G. (1985). *Creative problem solving: The Basic Course*. Buffalo, NY: Bearly Limited.

- Treffinger, D. J., & Isaksen, S. G. (2005). Creative problem solving: The history, development, and implications for gifted education and talent development. *Gifted Child Quarterly*, 49(4), 342-353.
- Treffinger, D. J., Isaksen, S. G., & Dorval, K. B. (2000). *Creative Problem Solving: An Introduction (3rd ed.)*. Waco, TX: Prufrock Press
- Urban, K. (1995). *Creativity-A component approach model*. Paper presented at the World Conference on the Education for the Gifted and Talented, Hong Kong.
- Wallas, G. (1926). *The art of thought*. London, UK: Jonathan Cape.
- Wang, Q., & Woo, H. L. (2007). Systematic planning for ICT integration in topic learning. *Journal of Educational Technology & Society*, 10(1), 148-156.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
- Weininger, O. (1977). Some thoughts on creativity and the classroom. *The Journal of Creative Behavior*.
- Weinberg, J. B., Engel, G. L., Gu, K., Karacal, C. S., Smith, S. R., White, W. W., & Yu, X. W. (2001). *A multidisciplinary model for using robotics in engineering education*. Paper presented at the Proceedings of the 2001 ASEE annual conference and exposition.
- Wells, J. (2015, October 23). *PIRPOSAL model: Design to understand(D2U)*. Paper presented at the Presentation at the Integrative STEM Education professional development for elementary education, STEM Education Collaboratory, Blacksburg, VA.
- Wilson, E. (2005). **통섭, 지식의 대통합** (최재천, 장대익, 역) In. 서울: 사이언스북스. (원저는 1998년 출간).
- Wing, J. (2011). Research notebook: Computational thinking – What and why. *The link magazine*, 6.
- 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*, 36(1881), 3717-3725.

Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & 소프트웨어itzer, M. (2011). Computational thinking and expository writing in the middle school. *ACM Transactions on Computing Education (TOCE)*, 11(2), 1-22.

Yakman, G. (2008). STΣ@M Education: an overview of creating a model of integrative education, PATT. In.

Ziman, J. M. (1980). *Teaching and learning about science and society*. Cambridge University Press.

## 부 록

- [부록 1] 내적 타당화를 위한 전문가 타당화 질문지
- [부록 2] 1차, 2차 설계원리 전반에 대한 전문가 타당화 결과
- [부록 3] 초기 개별 설계원리에 대한 1차 전문가 타당화 결과
- [부록 4] 2차 설계원리에 대한 전문가 타당화 결과
- [부록 5] 2차 설계원리의 개선을 위한 핵심 요소 도출
- [부록 6] 사용성 평가 질문지
- [부록 7] 사용성 평가 결과
- [부록 8] 창의적 문제해결력, 학업적 자기효능감 검사 문항
- [부록 9] 학생 면담 질문지
- [부록 10] 외적 타당화를 위한 교수·학습 과정안

[부록 1] 내적 타당화를 위한 전문가 타당화 질문지

“창의적 문제해결을 위한 교과-소프트웨어융합 수업설계  
모형개발”에 대한 전문가 타당도 검토

본 검토지는 창의적 문제해결을 위한 교과-소프트웨어융합설계 모형에 대한 타당화를 위해 제작되었습니다.

본 질문지는 ‘1. 연구 소개’ 부분과 ‘2. 타당도 검토’의 두 부분으로 구성되어 있으며, 연구 소개 부분은 1) 연구의 필요성과 목적 2) 구성요소, 설계 원리 및 지침, 모형의 개발 및 도출 과정과 주요 내용으로 구성되어 있습니다. 타당도 검토 부분은 1) 도출된 구성요소에 대한 타당화 질문지와 2) 수업설계 개념모형에 대한 타당화 질문지, 3) 설계원리 전반에 대한 타당화 질문지, 4) 개별 설계원리에 대한 타당화 질문지, 5) 수업설계의 절차모형에 대한 타당화 질문지로 구성되어 있습니다.

질문에 응답하는 과정에서 정확히 이해가 되지 않는 부분은 연구자에게 질문을 하실 수 있으며, 개방형 질문은 면담의 형식으로 진행할 수 있습니다. 전체 과정은 개방형 질문을 포함하여 약 30분에서 45분 정도 소요될 것으로 예상됩니다.

‘전문가 프로필’ 항목에 작성해주시는 ‘성명’은 자료식별용으로만 사용하고 연구의 익명성을 보장하기 위해 논문에는 언급되지 않을 것이며, 전문가임을 입증하기 위하여 필요한 내용, 즉 전공분야와 최종학력, 소속과 경력 부분만 논문에 제시될 것임을 말씀드립니다.

바쁘신 중에도 연구에 협조해 주셔서 대단히 감사합니다.

지현경 드림

서울대학교 교육학과 교육공학전공 박사과정

전문가 프로필

\* 성 명 :

\* 소속/직책 :

\* 전공분야 :

\* 주요 연구 분야 :

\* 최종학력 :

\* 실무 및 연구경력 :

### 1. 도출된 구성요소에 대한 타당화 질문지

- 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형 개발 연구의 구성요소는 관련 문헌 분석을 통해 도출하였습니다. 이에 도출된 구성요소가 타당한지를 알아보려고 합니다. 다음 설문지의 질문을 읽으시고 아래 평정척도에 따라 해당하는 곳에 체크하여 주시기 바랍니다.

(4: 매우 그렇다, 3: 그렇다, 2: 그렇지 않다, 1: 전혀 그렇지 않다)

평가문항		응답			
		1	2	3	4
선행문헌 탐색의 적절성	창의적 문제해결을 위한 교과-소프트웨어융합수업설계의 핵심 구성요소, 설계원리 및 지침, 개념모형, 절차모형을 도출하기 위해 관련 문헌의 탐색이 적절하게 이루어졌는가?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
선행문헌 고찰 결과 요약 및 해석의 적절성	선행연구는 창의적 문제해결을 위한 교과-소프트웨어융합수업설계의 핵심 구성요소, 설계원리 및 지침, 개념모형, 절차모형을 개발하기 위해 적절하게 요약되고 해석되었는가?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
구성요소의 적절성	도출된 구성요소는 창의적 문제해결을 위한 교과-소프트웨어융합수업설계 시 고려해야 할 핵심 요소들로 타당하게 구성되었는가?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	도출된 구성요소는 동일한 수준의 요소(또는 용어)로 적절하게 구성되었는가?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	각 구성요소별로 제시된 하위요소는 적절하게 구성되었는가?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
구성요소별 설명성	구성요소별 설명내용은 적절하게 표현되었는가?				
	· ‘교과융합 교육과정’에 대한 설명 내용	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	· ‘학습도구 및 정보자원’에 대한 설명내용	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	· ‘CT활용 창의적 융합설계활동’에 대한	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	설명 내용	
	· '공유 및 평가 지원'에 대한 설명내용	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

※ 수정 혹은 보완, 추가 개선 사항이 있으신 경우 자유롭게 작성해 주시기 바랍니다(특히, 3점 미만을 선택하신 경우 이에 대한 구체적 이유와 의견을 작성해 주시기 바랍니다)

## 2. 개념모형에 대한 타당성 검토

- 다음 설문 문항들은 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계의 개념 모형에 대한 타당성을 확인하는 목적을 지닙니다. 다음 설문지의 질문을 읽으시고 아래 평정척도에 따라 해당하는 곳에 √표 해주시기 바랍니다.

(4: 매우 그렇다, 3: 그렇다, 2: 그렇지 않다, 1: 전혀 그렇지 않다)

영역	평가문항	응답			
		1	2	3	4
타당성	본 개념 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계할 시 참고할 수 있는 설계모형으로 타당하다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
설명력	본 개념 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계할 시 따라야 할 개념들을 잘 설명하고 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
유용성	본 개념 모형은 창의적 문제해결을 위한 교과-	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	소프트웨어융합 수업을 설계하는데 유용하게 활용될 수 있다.				
보편성	본 개념 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계할 시 보편적으로 이용할 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
이해도	본 개념 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 이해하기 쉽게 표현되었다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

※ 수정 혹은 보완, 추가 개선 사항이 있으신 경우 자유롭게 작성해 주시기 바랍니다(특히, 3점 미만을 선택하신 경우 이에 대한 구체적 이유와 의견을 작성해 주시기 바랍니다)

### 3. 설계 원리 전반에 대한 타당화 질문지

- 다음 설문 문항들은 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 원리 전반에 대한 타당성을 확인하는 목적을 지닙니다. 다음 설문지의 질문을 읽으시고 아래 평정척도에 따라 해당하는 곳에 체크하여 주시기 바랍니다.

영역	평가문항	응답			
		1	2	3	4
타당성	본 설계원리는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 고려해야 할 원리로 타당하다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
설명력	본 설계원리는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 고려해야 할 원리를 잘	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	설명하고 나타나고 있다.				
유용성	본 설계원리는 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 유용하게 활용될 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
보편성	본 설계원리는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계 시 보편적으로 이용할 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
이해도	본 설계원리는 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 이해하기 쉽게 표현되었다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

※ 수정 혹은 보완, 추가 개선 사항이 있으신 경우 자유롭게 작성해 주시기 바랍니다(특히, 3점 미만을 선택하신 경우 이에 대한 구체적 이유와 의견을 작성해 주시기 바랍니다)

#### 4. 개별원리 전반에 대한 타당화 질문지 (생략)

#### 5. 절차모형에 대한 타당성 검토

- 다음 설문 문항들은 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 절차 모형에 대한 타당성을 확인하는 목적을 지닙니다. 다음 설문지의 질문을 읽으시고 아래 평정척도에 따라 해당하는 곳에 √표 해주시기 바랍니다.

(4: 매우 그렇다, 3: 그렇다, 2: 그렇지 않다, 1: 전혀 그렇지 않다)

영역	평가문항	응답
----	------	----

		1	2	3	4
타당성	본 절차 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계할 시 참고할 수 있는 설계모형으로 타당하다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
설명력	본 절차 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계할 시 따라야 할 절차들을 잘 설명하고 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
유용성	본 절차 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 유용하게 활용될 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
보편성	본 절차 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계할 시 보편적으로 이용할 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
이해도	본 절차 모형은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 이해하기 쉽게 표현되었다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

※ 수정 혹은 보완, 추가 개선 사항이 있으신 경우 자유롭게 작성해 주시기 바랍니다(특히, 3점 미만을 선택하신 경우 이에 대한 구체적 이유와 의견을 작성해 주시기 바랍니다)

타당성 검토에 참여해 주셔서 진심으로 감사드립니다.

[부록 2] 1차, 2차 설계원리 전반에 대한 전문가 타당화 결과

■ 1차 설계원리 전반에 대한 전문가 타당화

영역	전문가					평균	표준 편차	CVI	IRA
	A	B	C	D	E				
타당성	2	2	3	1	3	2.40	.548	0.40	
설명력	3	2	4	1	3	2.00	.000	0.60	
유용성	2	2	3	1	2	3.20	.447	0.20	0.00
보편성	2	2	3	1	3	1.20	.447	0.40	
이해도	3	2	3	2	2	2.60	.548	0.20	

■ 2차 설계원리 전반에 대한 전문가 타당화

영역	전문가					평균	표준 편차	CVI	IRA
	B	C	F	G	H				
타당성	4	4	4	3	4	3.80	.447	1.00	
설명력	4	4	4	3	3	3.60	.548	1.00	
유용성	4	4	4	3	3	3.60	.548	1.00	1.00
보편성	4	4	4	3	3	3.60	.548	1.00	
이해도	4	4	4	3	4	3.80	.447	1.00	

[부록 3] 초기 개별 설계원리에 대한 1차 전문가 타당화 결과

구성요소	원리	평균					평균	표준 편차	CVI	IRA
		A	B	C	D	E				
CT 활용 창의적 융합설계 활동	<b>1. 자기주도학습 제공의 원리</b>	2	2	3	1	3	2.20	.837	0.40	0.00
	1.1 친숙한 주제에 대한 토론과 발표를 통해 학습자가 스스로 문제를 설정, 계획하고 해결할 수 있도록 지원한다.	2	2	2	1	3	2.00	.707	0.20	
	1.2. 수집된 데이터 및 데이터의 패턴을 학생 스스로 분석하고 탐구할 수 있는 기회를 제공한다.	1	2	2	1	3	1.80	.837	0.20	
	<b>2. 감성적 체험 지원의 원리</b>	1	2	2	1	3	1.80	.837	0.20	
	2.1 학습자가 수업에 몰입하고 흥마와 동기를 가질 수 있도록 다양한 학습활동(경험, 관찰, 실습, 토의, 설계 등)을 구성한다.	2	2	2	1	3	2.00	.707	0.20	
	2.2 문제해결에 대한 동기 유발을 위해 다양한 발문과 자료를 활용한다.	3	2	2	1	3	2.20	.837	0.40	
	2.3 학습자의 성취와 새로운 문제에 대한 도전 의식을 촉진하기 위해 문제해결 이 후 내재적/외재적 보상체계를 마련한다.	2	2	3	1	3	2.20	.837	0.40	
	<b>3. 시각화 지원의 원리</b>	2	2	4	1	3	2.40	1.14	0.40	
	3.1 문제에 대한 해결 아이디어 및 사고과정을 순서대로 조직화 할 수 있도록 순서도 및 사고과	1	2	3	1	3	2.00	1.00	0.40	

	정 맵의 활용에 대한 내용을 안내한다.								
	3.2 학습 산출물이 시각적으로 표현될 수 있도록 온라인 플랫폼을 지원한다.	2	2	2	2	3	2.20	.447	0.20
	<b>4. 점진·반복성의 원리</b>	2	2	2	1	3	2.00	.707	0.20
	4.1 학습자가 점진적, 반복적으로 프로젝트를 설계하고 실행할 수 있도록 피드백을 제공한다.	2	2	2	1	3	2.00	.707	0.20
	4.2 창의적 융합설계 활동의 단계별로 발산적 사고와 수렴적 사고를 반복적으로 사용할 수 있도록 사고지원도구의 활용방법을 안내한다.	1	2	3	1	3	2.00	1.00	0.40
	<b>5. 맥락 친숙성의 원리</b>	2	2	3	1	3	2.20	.837	0.40
	5.1 일상생활에서 일어나는 친숙하고 실제적인 과제와 사례를 다양하게 제시한다.	2	2	3	1	3	2.20	.837	0.40
	5.2 시나리오 형태로 하나의 큰 주제 내에 발생할 수 있는 상황을 소재로 문제를 제시한다.	1	2	3	1	3	2.20	1.00	0.40
	5.3 실생활의 문제처럼 복잡하고 답이 하나 이상인 주제나 과제, 문제를 비구조적으로 진술한다.	2	2	4	1	3	2.40	1.14	0.40
	<b>6. 내용 융합의 원리</b>	1	2	2	2	3	2.00	.707	0.20
	6.1 교육과정 및 교과서의 내용 요소를 면밀히 검토하고 교과융합 요소를 탐색한다.	1	2	2	1	3	1.80	.837	0.20
	6.2 CT, 창의력, 문제해결력, 협업능력을 기반으로 통합 프로젝트형 주제를 선정한다.	2	2	2	1	3	2.00	.707	0.20

교과융합  
문제

	6.3 융합 하려는 교과들의 성취기준을 확인 및 분류하여 이를 중심으로 주제를 선정하고 프로젝트를 구성한다.	1	2	2	1	3	1.80	.837	0.20
	<b>7. 컴퓨팅 도구 지원의 원리</b>	2	2	2	1	3	2.00	.707	0.20
	7.1 초보 학습자도 쉽게 사용하고 산출물을 공유할 수 있는 직관적인 인터페이스의 컴퓨팅 도구(EPL, 파워포인트, 클라우드, 컴퓨터, 태블릿 등)를 선정한다.	2	2	3	1	3	2.20	.837	0.40
	7.2 학습자에게 흥미롭고 동기를 유발 할 수 있는 요소(직관적인 UI, 귀여운 이미지 등)가 포함된 컴퓨팅 도구를 제공한다.	3	2	3	1	3	2.40	.894	0.60
학습도구(창의적 사고도구, 컴퓨팅도구) 및 정보자원	7.3 컴퓨팅 도구를 간단하고 쉬운 수준(초보자용 도구)에서 점진적으로 복잡하고 복합적인 수준(전문가용 도구)으로 제공한다.	2	2	2	1	3	2.00	.707	0.20
	<b>8. 발산/수렴적 사고도구 지원의 원리</b>	1	2	3	1	3	2.00	1.00	0.40
	8.1 다양한 아이디어를 구상하고 도출할 수 있도록 발산적 사고도구(브레인스토밍, SCAMPER, 마인드맵 등)를 활용한다.	1	2	3	1	3	2.00	1.00	0.40
	8.2 문제에 대한 적절한 해결책을 찾을 수 있도록 수렴적 사고도구(HIT, PMI, ALU 등)를 활용한다.	1	2	3	1	3	2.00	1.00	0.40
	<b>9. 정보자원 및 시스템 지원의 원리</b>	2	2	2	2	3	2.20	.447	0.20

	9.1 학습내용과 학습 활동 지원을 위한 정보자원(프로젝트 관련 내용지식, 코딩명령어 카드 등)을 제공한다.	1	2	3	2	3	2.20	.837	0.40
	9.2 체계적인 학습자료 및 데이터 접근 방식에 대한 매뉴얼을 지원한다.	2	2	3	2	3	2.40	.548	0.40
	9.3 창의적 문제해결 활동 및 프로젝트 구현을 위한 온라인 학습공간 및 웹기반 시스템을 제공한다.	2	2	2	2	3	2.20	.447	0.20
	<b>10. 협력적 상호작용의 원리</b>	1	2	3	1	3	2.00	1.00	0.40
	10.1 문제해결이 협력적으로 이루어 질 수 있도록 다양한 특성을 가진 구성원의 팀을 구성하여 그룹활동을 촉진한다.	2	2	3	1	3	2.20	.837	0.40
개방적 학습환경	10.2 학습자들 간 상호 지원활동이 지속적으로 이루어지고 있는지 점검하고 적응적인 피드백을 제공한다.	2	2	2	1	3	2.00	.707	0.20
	<b>11. 개방적 학습환경 제공의 원리</b>	3	2	2	1	3	2.20	.837	0.40
	11.1 학습자의 자유로운 사고를 촉진하고 학습자간 의견을 부담없이 공유할 수 있는 열린 학습 분위기를 형성한다.	3	2	4	1	3	2.60	1.14	0.60
	<b>12. 산출물 공유 및 평가의 원리</b>	2	2	3	1	3	2.20	.837	0.40
공유 및 평가	12.1 창의적 문제해결 과정에 대한 평가기준을 공유한다.	2	2	4	1	3	2.40	1.14	0.40

	12.2 프레젠테이션을 통해 산출물을 시연하고 의견을 나눌 수 있도록 안내한다.	2	2	3	1	3	2.20	.837	0.40
	12.3 학습자의 산출물을 다양한 관점에서 객관적으로 평가할 수 있도록 모듈 간/모듈 내 동료 평가 및 자기평가 기회를 제공한다.	2	2	3	1	3	2.20	.837	0.40
	<b>13. 안내 제공의 원리</b>	3	2	2	1	3	2.20	.837	0.40
	13.1 학습자가 수업을 하는 과정에서 진행할 내용 및 운영방식을 사전에 안내한다.	3	2	3	1	3	2.40	.894	0.60
안내	13.2 학습자에게 활용 도구(프로그래밍 언어, 수렴·발산적 사고도구)에 대한 사전 안내를 실시한다.	3	2	3	1	3	2.40	.894	0.60
	13.3 학습 과제 및 활동에 대해 분명한 준거(과제 수행 시간, 완성도, 평가 계획)를 확립하고 학습자가 항상 열어볼 수 있도록 안내문, 플랫폼 등을 통해 구체적으로 제시한다.	2	2	3	1	3	2.20	.837	0.40

[부록 4] 2차 설계원리에 대한 전문가 타당화 결과

구성요소	원리	전문가					평균	표준 편차	CVI	IR A
		B	C	F	G	H				
교과융합 교육과정	<b>1. 적절성 확인의 원리</b>	4	4	4	3	3	3.60	.548	1.00	0.7
	1.1 융합하려는 교과 및 단원의 성취기준, 학습내용, 교수학습 방법, 평가방법, 학습자 수준을 확인하고 교과융합소프트웨어 수업이 교과융합 소프트웨어 수업으로 구현할 수 있는지를 판단한다.	4	3	4	3	2	3.20	.837	0.80	
	1.2 학습 환경, 자원, 학교 방침 및 분위기, 전문가 그룹 등 수업을 위한 제반 자원과 환경을 파악하고 교과융합 소프트웨어수업의 실행 및 지원 가능성을 확인한다.	4	3	4	3	4	3.60	.548	1.00	
	<b>2. 융합 문제 설계의 원리</b>	4	4	4	3	3	3.60	.548	1.00	
	2.1 융합하려는 교과목의 성취기준, 학습내용, 소프트웨어수업 요소와의 연계 요소와 학습자 수준을 고려하여 공통주제 수업 목표, 과제 형식을 선정한다.	4	3	4	3	4	3.60	.548	1.00	
	2.2 실생활의 상황을 바탕으로 학습자가 친숙함을 느낄 수 있는 시나리오 형태로 문제를 제시한다.	4	4	4	3	4	3.80	.447	1.00	
	2.3 해결안이 하나 이상으로 제시될 수 있	4	4	4	3	4	3.80	.447	1.00	

	는 복합적이고 비구조화된 형태의 문제를 설계한다.									
	<b>3. 수업 재구성의 원리</b>	4	4	4	3	3	3.60	.548	1.00	
	3.1 교과 특성, 주제의 연계성, 학습자 수준을 고려하여 단원의 순서를 바꾸거나 내용을 생략하고 주제와 관련된 주요 내용을 중심으로 수업을 재구성한다.	4	4	4	3	4	3.80	.447	1.00	1.0
	3.2 융합 주제를 중심으로 융합 할 교과와 소프트웨어 수업의 내용이 하나의 맥락으로 통합될 수 있도록 수업을 재구성한다.	4	3	4	3	4	3.40	.548	1.00	
	<b>4. 사용 용이성의 원리</b>	4	4	4	3	3	3.60	.548	1.00	
	4.1 초보 학습자도 쉽게 사용할 수 있는 직관적 인터페이스의 프로그래밍 소프트웨어 도구를 선정한다.	4	4	2	3	4	3.80	.447	0.80	0.7
	4.2 학습자들이 쉽게 이해하고 사용할 수 있는 사고 지원 도구를 선정한다.	4	3	4	3	4	3.40	.548	1.00	
학습도구 및 정보자원	<b>5. 정보자원 제공의 원리</b>	4	4	4	3	4	3.80	.447	1.00	
	5.1 창의적 문제해결 활동 및 프로젝트 구현을 위한 온라인 학습공간 및 웹기반 시스템을 제공한다..	4	3	4	3	3	3.20	.447	1.00	0.8
	5.2 학습자료의 검색을 효율적으로 수행할 수 있도록 학습 내용과 관련한 검색 키워드 및	4	3	4	3	4	3.40	.548	1.00	

	참고사이트, 자료 아카이브를 사전에 만들어 제공한다.											
	5.3 자료를 정리/조직하고 모둠원간 공유할 수 있도록 온라인 공동 작업 소프트웨어 프로그램을 활용한다.	4	4	4	3	4	3.80	.447	1.00			
	5.4 학습 과제 및 활동에 대해 분명한 증거(과제 수행 시간, 완성도, 평가 계획)에 대해 안내문을 작성하여 온라인 학습공간에 게시한다.	4	3	4	3	2	3.00	.707	0.80			
	<b>6. 문제이해 지원의 원리</b>	4	4	4	3	4	3.80	.447	1.00			
	6.1 수업 내용과 관련된 실생활의 예시 자료(동영상, 애니메이션, 신문기사 등)을 제시하고 학습자의 경험을 공유하도록 안내한다.	4	3	4	3	4	3.40	.548	1.00			
창의적 교과-소프트웨어 융합 설계활동 지원	6.2 학습자가 문제 상황을 이해하고 정의할 수 있도록 수업 주제, 해결해야 하는 문제, 해결 목적과 관련한 구체적인 질문을 제시한다.	4	4	4	3	4	3.80	.447	1.00	1.0		
	6.3 학습자 수준을 고려하여 최종적으로 도출할 수 있는 산출물의 예시를 제시한다.	4	3	4	3	4	3.60	.548	1.00			
	<b>7. 아이디어 생성 지원의 원리</b>	4	4	4	3	3	3.60	.548	1.00			
	7.1 문제에 대한 다양한 해결 아이디어를 도출할 수 있도록 발산적 사고도구(브레인스토	4	3	4	3	4	3.60	.548	1.00	1.0		

밍, 속성열거법, SCAMPER)를 활용한다.									
7.2 문제에 대한 해결책을 찾을 수 있도록 수렴적 사고도구(HIT, PMI, 평가행렬법)를 활용한다.	4	3	4	3	4	3.60	.548	1.00	
7.3 아이디어의 정리와 조직화를 위해 시각화 지원도구(종이, 포스트잇, 카드보드, 화이트보드)를 활용한다.	4	3	4	3	4	3.60	.548	1.00	
<b>8. 논리화 지원의 원리</b>	4	4	4	3	4	3.80	.447	1.00	
8.1 중요한 아이디어를 선별하여 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호작용 등을 이해하고 해결안을 표현할 수 있도록 스토리보드 형태의 활동지를 제공한다.	4	3	4	3	4	3.60	.548	1.00	1.0
8.2 알고리즘 작성 지원을 위한 학습자료(코딩 명령어 매뉴얼, 코딩 작성 예시자료)를 제시한다.	4	3	4	3	4	3.60	.548	1.00	
8.3 의사코드와 순서도의 작성 방법을 안내하고, 최종 산출물의 구현 가능성을 모듈별 논의를 통해 검토할 수 있도록 안내한다.	4	4	4	3	4	3.80	.447	1.00	
<b>9. 점진/반복적 문제해결 지원의 원리</b>	4	3	4	3	4	3.60	.548	1.00	
9.1 학습자가 점진적, 반복적으로 프로그램을 테스트하고 디버깅할 수 있도록 피드백을	4	4	4	3	4	3.80	.447	1.00	1.0

제공한다.									
9.2 학습자의 리믹스 활동을 지원하기 위해 프로그래밍의 구성요소를 일부 제공하거나, 동료나 교수자 및 온라인상의 다른 학습자들이 제작한 소프트웨어의 코드를 열람하고 이를 바탕으로 알고리즘을 발전시킬 수 있도록 안내한다.	4	3	4	3	4	3.60	.548	1.00	
9.3 컴퓨팅 사고력과 관련한 지식(반복, 연산자, 변수, 알고리즘 등)을 활용하여 점차적으로 복잡하고 어려운 코드를 작성할 수 있도록 난이도를 고려하여 수업을 설계한다.	4	3	4	3	4	3.60	.548	1.00	
<b>10. 동료 프로그래밍 지원의 원리</b>	4	4	4	3	3	3.60	.548	1.00	
10.1 컴퓨터를 활용한 프로그래밍 작업 경험이 풍부한 학습자들과 상대적으로 부족한 학습자들을 하나의 모듈으로 구성한다.	4	3	4	3	4	3.60	.548	1.00	
10.2 운전자 역할의 학습자와 네비게이터 역할의 학습자가 동시적·협력적으로 작업할 수 있도록 모듈별로 역할을 배정하고 각자의 역할에 대해 설명한다.	4	4	4	3	3	3.60	.548	1.00	1.0
10.3 일정 기준을 중심으로 운전자와 네비게이터의 역할을 바꾸어 서로의 작업을 검토하고 보완할 수 있도록 안내한다.	4	4	4	3	4	3.80	.447	1.00	

	<b>11. 공유 및 평가 지원의 원리</b>	4	4	4	3	4	3.80	.447	1.00	
	11.1 학습자가 최종 산출물을 공유하고 배 포할 수 있도록 공유 방법을 안내한다.	4	3	4	3	4	3.60	.548	1.00	
	11.2 모둠 별 발표를 통해 산출물을 시연하 고 의견을 나눌 수 있는 기회를 제공한다.	4	4	4	3	4	3.80	.447	1.00	
공유 및 평가 지원	11.3 학습자의 산출물을 다양한 관점에서 객관적으로 평가할 수 있도록 교수자 및 모둠 별 평가 기회를 제공한다.	4	4	4	3	3	3.60	.548	1.00	0.8
	11.4 학습자가 전체적인 학습활동을 돌이켜 볼 수 있도록 구조화된 성찰일지 작성을 지원 한다.	4	4	4	3	4	3.80	.447	1.00	
	11.5 학습자의 자유로운 사고를 촉진하고 학습자간 의견을 부담없이 공유할 수 있는 열 린 학습 분위기를 형성한다.	4	4	4	3	2	3.40	.894	0.80	

[부록 5] 2차 설계원리의 개선을 위한 핵심 요소 도출

구분	Isaksen, Doral, & Treffinger (2000)	전용주 (2018)	함성진, 김순화, 송기상 (2014)	홍미영 (2012)	공통내용 (2차 원리에 적용)	구성요소	
창의적 문제해결단계	기회의 구성	문제인식 및 분석	문제상황 제시	과제 파악	문제 이해	창의적 교과-소프트웨어 융합 설계활동 지원	
	자료의 탐색		자료수집 및 분석	과정설계			
	문제의 골격구성	아이디어 구상	문제진술	문제확인			
	아이디어의 생성		창의적 융합설계	자료탐색			
	해결책의 개발	설계	제작 및 시뮬레이션	아이디어 생성	아이디어 생성		논리화 점진/반복적 문제해결
	수용도대의 구축	구현 및 적용	감성적 체험 및 평가	실행조건 확립	공유 및 평가		

[부록 6] 사용성 평가 질문지

1. 개념모형과 절차모형에 대한 사용성 평가

다음 설문 문항들은 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계를 위한 모형과 원리의 사용성에 대한 평가를 위한 문항입니다. 원리와 모형의 타당성을 묻는 것입니다. 다음 설문지의 질문을 읽으시고 4단계 평정척도에 따라 해당하는 곳에 √표 해주시기 바랍니다.

(4: 매우 타당하다, 3: 타당하다, 2:타당하지 않다, 1: 전혀 타당하지 않다)

문항	응답			
	4	3	2	1
개념모형과 그 설명은 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계가 무엇인지 이해하는데 도움이 된다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
개념모형과 그 설명은 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계를 위하여 무엇을 해야 하는지(어떤 요소들을 고려해야 하는지) 이해하는데 도움이 된다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
절차모형과 그 설명은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하기 위해 어떠한 단계를 따라야 하는지를 잘 설명하고 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
절차모형과 그 설명은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 도움이 된다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
설계원리 및 상세지침은 창의적 문제해결을 위한 교과-소프트웨어융합 수업을 설계하는데 도움이 된다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
기타의 견				

■ 이 연구에서 제시한 개념모형의 강점과 약점은 무엇입니까?

강점	약점

■ 이 연구에서 제시한 개념모형 중 중점적으로 개선되어야 할 부분은 무엇입니까?

--

■ 이 연구에서 제시한 절차모형의 강점과 약점은 무엇입니까?

강점	약점

--	--

- 이 연구에서 제시한 절차모형 중 중점적으로 개선되어야 할 부분은 무엇입니까?

--

## 2. 개별원리 전반에 대한 사용성 평가(생략)

- 이 연구에서 제시한 설계원리 및 상세지침의 강점과 약점은 무엇입니까?

강점	약점

- 이 연구에서 제시한 설계원리 및 상세지침 중 중점적으로 개선되어야 할 부분은 무엇입니까?

- 설계하기 어렵거나 이해하기 어려운 부분은 무엇입니까?

- 현재의 설계모형과 원리에서 추가되어야 할 사항은 무엇입니까?

사용성 평가에 참여해 주셔서 감사합니다.

[부록 7] 사용성 평가 결과

구성 요소	설계원리 및 상세지침	현장 교사 평가			평균	CVI	IRA		
		A	B	C					
교과-소프트웨어융합 교육과정	1. 수업 실행성의 원리	4	4	3	3.67	1.00	0.8		
	1.1	4	4	4	4.00	1.00			
	1.2	4	3	3	3.00	1.00			
	1.3	4	3	3	3.00	1.00			
	1.4	2	3	4	3.00	0.68	1.0		
	2. 맥락 재구성의 원리	4	4	4	4.00	1.00			
	2.1	4	4	4	4.00	1.00			
	2.2	4	4	4	4.00	1.00			
	3.	4	3	4	3.67	1.00			
	3.1	4	4	4	4.00	1.00			
	3.2	3	3	4	3.33	1.00			
	3.3	3	4	3	3.33	1.00			
	소프트웨어 학습도구	4. 실제적 문제의 원리	4	4	4	4.00		1.00	1.0
		4.1	4	4	4	4.00		1.00	
4.2		4	4	4	4.00	1.00			
5.		4	4	4	4.00	1.00	1.0		
5.1		4	3	4	3.67	1.00			
5.2		4	3	4	3.67	1.00			
교과-소프트웨어융합 학습지원	6. 소프트웨어도구 적합성의 원리	4	3	4	3.67	1.00	1.0		
	6.1	4	4	3	3.67	1.00			
	6.2	4	3	4	3.67	1.00			
	6.3	4	3	4	3.67	1.00			
	7. 프로그램 모듈 구성의 원리	4	3	4	3.67	1.00	0.8		
	7.1	4	3	4	3.67	1.00			
	7.2	4	3	4	3.67	1.00			
교과-소프트웨어융합 수업활동	7.3	4	2	4	3.33	0.68	0.5		
	8. 문제이해 지원의 원리	4	3	4	3.67	1.00			
	8.1	4	3	4	3.67	1.00			
	8.2	4	2	4	3.33	0.68			
	8.3	4	2	4	3.33	0.68			
	9. 아이디어 생성 지원성의 원리	4	4	4	4.00	1.00	1.0		
9.1	3	4	4	3.67	1.00				

	9.2	3	4	4	3.67	1.00	
	9.3	4	4	4	4.00	1.00	
	10. 논리화 지원성의 원리	4	4	4	4.00	1.00	
	10.1	4	3	4	3.67	1.00	0.7
	10.2	4	2	4	3.33	0.68	
	11. 점진적 문제해결 지원성의 원리.	4	3	4	3.67	1.00	
	11.1	4	2	4	3.33	0.68	0.5
	11.2	4	3	4	3.67	1.00	
	11.3	4	2	4	3.33	0.68	
공유 및 평가	12. 공유 및 평가 지원성의 원리	4	4	3	3.67	1.00	
	12.1	4	4	4	4.00	1.00	1.0
	12.2	4	4	3	3.67	1.00	
	12.3	4	4	4	4.00	1.00	

## 창의적 문제해결을 위한 교과-소프트웨어융합 수업설계 모형 개발 연구에 대한 설문

안녕하십니까? 본 설문지는 창의적 문제해결을 위한 교과-소프트웨어융합 수업 설계의 원리와 모형을 실제 수업에 적용하여 그 효과를 조사하는 데 목적이 있습니다. 코딩 프로그램을 활용하여 진행한 수업의 효과를 조사하는 것이 본 설문의 목적입니다.

이 설문지는 창의적 문제해결을 위한 교과-소프트웨어융합 수업에 대한 학생들의 의견을 받고자 작성되었습니다. 이 설문지를 통해 수집된 의견은 창의적 문제해결을 위한 교과-소프트웨어융합 수업모형 개발의 기초자료로 활용될 것입니다. 본 설문조사는 **약 15분** 정도가 소요되며, 본 설문에 참여해 주신 분 중 몇 분에게는 면담을 요청할 수 있습니다. 설문조사를 참여하거나 참여하지 않음에 따라 발생하는 손실이나 이득은 없습니다. 이 연구를 통해 얻는 개인 정보는 사전 부여받은 번호뿐이며, 모든 개인정보의 비밀 보장을 위해 최선을 다할 것입니다. 또한, 설문 결과는 성적에 반영되지 않으며, 연구 이외의 목적으로 사용하지 않을 것을 약속드립니다. 귀하가 이 질문지에 응답하는 것은 이러한 사항에 대하여 사전에 알고 있었으며 연구 참여를 허용한다는 동의로 간주될 것입니다. 만약 설문조사를 마쳤더라도 본인의 설문조사 결과가 본 연구에 이용되기를 원하지 않을 경우, 언제든지 연구담당자에게 요청하여 연구 참여에 대한 동의를 철회할 수 있습니다.

이 연구와 관련하여 문의사항이 있으시면 아래의 연락처로 연락해주시기 바랍니다. 바쁘신 중에도 적극적으로 협조해 주셔서 대단히 감사합니다.

	연구담당자: 서울대학교 교육학과 지현경
	연락처: 000-0000-0000

이 연구에 참여하기를 원하십니까?    예     아니오

※ 다음의 질문을 읽고, 답해주세요.

성별( 남  / 여  )

학년(                      학년) 번호(                      번)

**A. 창의적 문제해결력**

다음 문항들은 **여러분이 얼마나 창의적으로 문제를 생각하고 해결 하고 있는지를 알아보기 위한 것**입니다. 아래 내용을 주의 깊게 읽고, 자신의 생각과 가장 일치하는 번호를 선택해 주세요. 정답이 있거나 좋고 나쁜 답이 있는 것이 아니므로 솔직하게 자신의 생각을 표시해 주세요.

※ 자신의 생각과 가장 일치하는 곳에 V를 표시해 주세요.

문항	전혀 그렇지 않다	그렇지 않다	보통이다	그렇다	매우 그렇다
1 나는 수업 시간 중 많은 것에 호기심을 가지고 계속 질문한다.	①	②	③	④	⑤
2 나는 주어진 문제에 대하여 다양한 해답을 찾아 내며, 이따금 독특한 해답을 제시한다.	①	②	③	④	⑤
3 나는 수업시간에 의사를 자유로이 표현하며, 이따금 의견이 맞지 않을 때는 과격하게 맞서거나, 고집을 부린다.	①	②	③	④	⑤
4 나는 평소에 유머가 풍부하며, 남이 우습지 않은 상황에서도 남들을 곤잘 웃긴다.	①	②	③	④	⑤
5 나는 공부시간에 머리를 쓰는 놀이를 좋아한다.	①	②	③	④	⑤
6 나는 참신하고 남다른 생각을 말할 수 있다.	①	②	③	④	⑤
7 나는 이미 알려진 것과는 다른 새로운 방법으로 문제를 풀 수 있다.	①	②	③	④	⑤
8 내가 만든 것은 새로워서 다른 친구들이 만든 것과는 많이 다르다.	①	②	③	④	⑤

9	나는 문제를 풀어낼 아이디어를 다양하고 풍부하게 만들어 낸다.	①	②	③	④	⑤
10	나는 서로 상관없이 보이는 것을 잘 연결 짓는다.	①	②	③	④	⑤
11	나는 실제로 있는 사실과 상상을 구별할 줄 안다.	①	②	③	④	⑤
12	나는 수업시간에 아이디어나 결론을 꼼꼼하고 찬찬히 다음으로 이어나갈 수 있다.	①	②	③	④	⑤
13	나는 공부시간에 말이 맞는 말인지 또는 틀린 말인지를 판단할 줄 안다.	①	②	③	④	⑤
14	나는 친구들과 다양한 정보를 바탕으로 혼자서 결론을 이끌어 낼 수 있다.	①	②	③	④	⑤
15	나는 주어진 문제와 관계가 있는 정보를 찾아낼 수 있다.	①	②	③	④	⑤
16	나는 어렵고 힘든 것도 쉽게 포기하지 않고 끝까지 하려고 한다.	①	②	③	④	⑤
17	나는 이 과목의 다른 주제에 대해서도 더 알고 싶다.	①	②	③	④	⑤
18	나는 수업시간의 공부 내용이 매우 재밌다.	①	②	③	④	⑤
19	나는 목표에 달성하지 못했다고 생각되면 목표달성을 위해 더 노력한다.	①	②	③	④	⑤
20	나는 목표를 이루었다고 생각하면 그 다음 단계의 목표를 정한다.	①	②	③	④	⑤

---

## B. 학업적 자기효능감

---

다음 문항들은 **여러분이 수업을 듣고 적절한 수업활동을 할 수 있는지에 대한 여러분의 생각을 알아보기 위한 것**입니다. 아래 내용을 주의 깊게 읽고, 자신의 생각과 가장 일치하는 번호를 선택해 주세요. 정답이 있거나 좋고 나쁜 답이 있는 것이 아니므로 솔직하게 자신의 생각을 표시해 주세요.

※ 자신의 생각과 가장 일치하는 곳에 V를 표시해 주세요.

문항	전혀 그렇지 않다	그렇지 않다	보통이다	그렇다	매우 그렇다
1 나는 복잡하고 어려운 문제에 도전하는 것이 재미있다.	①	②	③	④	⑤
2 가능하다면 어려운 과목은 피해가고 싶다.	①	②	③	④	⑤
3 나는 깊이 생각해야 하는 문제보다는 쉽게 풀 수 있는 문제를 더 좋아한다.	①	②	③	④	⑤
4 비록 실패하더라도 다른 친구들이 풀지 못한 문제에 도전하는 것이 즐겁다.	①	②	③	④	⑤
5 시간이 많이 들더라도 깊이 생각하게 만드는 과목이 더 재미있다.	①	②	③	④	⑤
6 만약 여러 과목 중 몇 과목만을 선택할 수 있다면, 쉬운 과목만을 선택할 것이다.	①	②	③	④	⑤
7 학교 공부는 무조건 쉬울수록 좋다.	①	②	③	④	⑤
8 쉬운 문제보다는 조금 틀리더라도 어려운 문제를 푸는 것이 더 좋다.	①	②	③	④	⑤
9 나는 쉬운 문제를 여러 개 푸는 것 보다 어려운 문제를 하나 푸는 것을 더 좋아한다.	①	②	③	④	⑤
10 나는 쉬운 과목보다는 어려운 과목을 좋아한다.	①	②	③	④	⑤
11 나는 수업시간에 새로 배운 것들을 이미 알고 있는 것과 쉽게 연결시킬 수 있다.	①	②	③	④	⑤
12 나는 보통 공부를 시작하기 전에 계획을 세우고, 거기에 맞추어 공부한다.	①	②	③	④	⑤
13 나는 수업시간 중에 중요한 내용을 잘 기록할 수 있다.	①	②	③	④	⑤

14	내가 싫어하는 수업시간에도 주의집중을 잘 할 수 있다.	①	②	③	④	⑤
15	나는 복잡하고 어려운 내용을 기억하기 쉽게 바꿀 수 있다.	①	②	③	④	⑤
16	나는 수업시간에 배운 내용을 잘 기억할 수 있다.	①	②	③	④	⑤
17	나는 어떻게 공부하는 것이 효과적인 방법인지를 잘 안다.	①	②	③	④	⑤
18	나는 정해진 시간 안에 주어진 과제를 잘 마칠 수 있다.	①	②	③	④	⑤
19	나는 수업시간에 배운 내용 중 내가 무엇을 알고, 무엇을 모르는지 정확히 판단할 수 있다.	①	②	③	④	⑤
20	나는 수업시간에 배운 내용 중 중요한 것이 무엇인지를 잘 파악할 수 있다.	①	②	③	④	⑤
21	선생님과 친구들 앞에서 발표하는 것은 내게 너무 큰 스트레스를 준다.	①	②	③	④	⑤
22	시험을 치루기 전에는 시험을 망칠 것 같은 생각이 든다.	①	②	③	④	⑤
23	수업시간 중에 선생님이 문제를 풀라고 시킬까봐 불안하다.	①	②	③	④	⑤
24	토론을 할 때, 혹시 창피를 당할까봐 내 의견을 제대로 발표하지 못한다.	①	②	③	④	⑤
25	수업시간에 발표를 할 때, 실수를 할 것 같아 불안하다.	①	②	③	④	⑤
26	선생님이 모두에게 질문을 할 때, 답을 알아도 대답하지 못한다.	①	②	③	④	⑤
27	시험이 다가오면 불안해서 잠을 이룰 수가 없다.	①	②	③	④	⑤
28	나는 시험 때만 되면 우울해진다.	①	②	③	④	⑤

- 끝 -

[부록 9] 학생 면담 질문지

창의적 문제해결을 위한  
교과-소프트웨어융합 수업설계 모형 개발 연구

학생 면담 질문지

1. 과거 학교 수업에서 다른 교과를 배울 때 소프트웨어를 활용하여 공부한 경험이 있나요?
  - 1-1. 만약 경험이 있다면, 언제 경험했고 그 수업에 대해 어떻게 생각하나요?
  - 1-2. 그 당시의 수업과 지금 수업을 비교할 때 차이점은 무엇인가요?
2. 다른 교과와 소프트웨어를 융합하여 공부하는 것의 장점은 무엇이라고 생각하나요?
  - 2-1. 교과-소프트웨어 융합 수업의 어떤 학습활동이 도움이 되었나요?
  - 2-2. 모듈원이나 짝 끼리의 협력이 잘 이루어졌나요?
3. 다른 교과와 소프트웨어를 융합하여 공부할 때 어려웠던 점은 무엇인가요?
  - 3-1. 수업 전 코딩 소프트웨어에 대해 충분히 배울 수 있었나요?
  - 3-2. 컴퓨터를 다루고 코딩 소프트웨어를 활용하는 방법이 어려웠나요?
  - 3-3. 문제에 대해 새로운 아이디어를 생각하고, 중요한 아이디어의 알고리즘을 생각해 보는 활동에 어려움은 없었나요?
  - 3-4. 코딩 소프트웨어를 활용하여 생각한 아이디어를 구현해 보는 활동에 어려움은 없었나요?
  - 3-5. 친구들의 결과물에 대해 서로 토론하고, 결과물을 다시 활용하여 새로운 작업을 하는 것에 있어서 어려움은 없었나요?
  - 3-6. 개선하면 좋을 만한 부분은 무엇이 있나요?
4. 교과-소프트웨어 융합 수업에서 가장 기억에 남는 활동은 무엇인가요?
  - 4-1. 그 활동이 가장 기억에 남는 이유는 무엇인가요?
  - 4-2. 새로운 아이디어를 생각하고 코딩 소프트웨어를 활용하여 문제를 해결

하는 활동을 하면서 배운 점이나 느낀 점은 무엇인가요?

**5. 사회나 국어 과목 이외의 다른 과목에서도 소프트웨어를 융합하여 활용하는 수업을 받고 싶나요?**

5-1. 추가적으로 소프트웨어와 융합하여 배우고 싶은 과목과 그 이유는 무엇인가요?

[부록 10] 외적 타당화를 위한 교수·학습 과정안

I. 차시별 지도계획

구분	차시	학습내용	교과	단원	비고
문제 발견 및 이해	1/6	<ul style="list-style-type: none"> <li>일상생활 속에서 찾아볼 수 있는 대도시의 환경문제를 알아보고, 생태도시의 필요성과 이를 알리기 위한 노력에 대해 이해할 수 있다.</li> <li>생태도시를 다루는 다양한 매체 자료를 수집하고, 내가 계획한 생태도시 홍보 애니메이션의 주제를 찾을 수 있다.</li> </ul>	사회 국어	사회 3. 지속가능한 지구촌 국어 4. 효과적으로 발표해요	
아이디어 생성	2/6	<ul style="list-style-type: none"> <li>내가 만들고 싶은 홍보 애니메이션에 대한 아이디어를 생성하고 구상할 수 있다.</li> </ul>	사회 국어	사회 3. 지속가능한 지구촌 국어 4. 효과적으로 발표해요	
논리화	3/6	<ul style="list-style-type: none"> <li>핵심 아이디어를 선별하여 홍보 애니메이션을 제작하기 위한 논리적인 해결안(알고리즘)을 작성할 수 있다.</li> </ul>	실과	소프트웨어교육	
문제해결	4/6	<ul style="list-style-type: none"> <li>설계한 알고리즘을 바탕으로 EPL을 활용한 홍보 애니메이션을 제작할 수 있다.</li> </ul>	실과	소프트웨어교육	
	5/6		실과	소프트웨어교육	
공유 및 발표	6/6	<ul style="list-style-type: none"> <li>학습활동의 산출물을 온라인 플랫폼에 공유하고 발표한다.</li> </ul>	사회 국어	사회 3. 지속가능한 지구촌 국어 4. 효과적으로 발표해요	

## II. 차시별 수업계획

주 제	환경문제를 알리는 홍보자료(엔트리) 만들기	차 시	1~2/6	수업 모형	
학습 목표	일상생활에서 찾아볼 수 있는 환경문제를 해결하기 위한 노력을 하고 있는 다양한 생태도시를 찾아보고, 내가 계획한 생태도시의 홍보자료를 계획할 수 있다.				
자 료	교 사	교사용 컴퓨터			
	학 생	크롬북, 활동지, 패들렛			

단계	활동 요소	교수 · 학습 활동	자료(☒) 및 유의점(☉)
상황 제시	동기 유발	<p>■ 동기유발 및 학습문제 확인</p> <ul style="list-style-type: none"> <li>- 중국으로부터의 미세먼지가 한국에 미치는 영향에 대한 영상을 시청 후 이야기해 보기 <ul style="list-style-type: none"> <li>▪ 영상에서는 어떤 문제를 이야기하고 있나요??</li> <li>▪ 미세먼지 외에 우리나라가 다른 나라로부터 영향을 받는 환경문제가 있나요?(아마존 삼림 파괴, 지구온난화 등)</li> <li>▪ 지구촌의 다양한 나라와 도시들의 환경문제 해결이 몇몇 나라의 노력으로 가능할까요?(생태도시 계획)</li> <li>▪ 그렇다면 우리는 어떤 노력이 필요할까요?</li> <li>▪ 이러한 노력을 알리기 위해 어떤 활동을 할 수 있을까요?(홍보)</li> </ul> </li> <li>- 생태도시의 중요성을 생각해보고 직접 계획한 생태도시를 알리기 위한 노력을 이야기해 보기 <ul style="list-style-type: none"> <li>▪ 생태도시 꾸리찌바는 환경문제 해결을 위해 어떤 노력을 했나요?</li> <li>▪ 책을 읽고 직접 생태도시를 계획해 보면서 생태도시를 가꾸는 것의 중요성을 알 수 있었나요?</li> <li>▪ 내가 계획한 생태도시의 특징과 중요성을 알리기 위해 어떤 노력을 할 수 있을까요?</li> </ul> </li> </ul>	<p>☒ 환경문제 홍보 동영상</p> <p>☒ 도서 ‘숨쉬는 도시 꾸리찌바’</p>
정보 수집	문제 해결을	<p>■ 정보 수집 및 분석하기</p> <ul style="list-style-type: none"> <li>- 다양한 생태도시들과 내가 계획한 생태도시를 비교했</li> </ul>	☒ 모듈별

단계	활동 요소	교수 · 학습 활동	자료(☞) 및 유의점(☉)
	위한 정보 수집하기	<p>을 때 같은 점과 다른 점에 대해 자료를 수집하기</p> <ul style="list-style-type: none"> <li>▪ 생태도시들의 환경문제 해결에 대한 노력을 이슈별로 분류하여 검색 키워드를 제공하기 (예: 미세먼지, 쓰레기 처리문제, 자동차 매연 등)</li> <li>▪ 수집한 자료를 패들릿에 게시하기</li> </ul> <p>- 모둠별로 수집한 자료를 분석하기</p> <ul style="list-style-type: none"> <li>▪ 수집한 자료를 비슷한 것끼리 분류하기</li> <li>▪ 우리 모둠이 활용할 주제 선택하기</li> </ul>	패들릿
문제 정의	문제를 이해하고 정의하기	<ul style="list-style-type: none"> <li>■ 모둠별 홍보자료 주제 정하기</li> <li>- 다양한 생태도시들의 캠페인 및 홍보자료를 시청하기</li> <li>- 엔트리를 활용해 다른 사용자들이 제작한 애니메이션을 시청하고, 최종 산출물의 형태를 이해하기</li> <li>- 모둠별로 수집, 분석한 자료를 토대로 엔트리를 활용하여 내가 만든 생태도시를 알리기 위해 어떤 홍보물을 만들지 정하기</li> </ul>	
정리	아이디어 생성	<ul style="list-style-type: none"> <li>■ 모둠별로 제작하고자 하는 홍보자료 대한 아이디어를 생각하기(브레인스토밍 활동)</li> <li>- 모둠별 제작하고자 하는 홍보자료의 내용과 특징에 대해 아이디어를 생각하고 논의하기</li> <li>- 숨쉬는 도시 꾸리지마 책을 참고해서 홍보자료에 대한 다양한 아이디어 떠올려보기 <ul style="list-style-type: none"> <li>▪ 우리도시와 다른 도시의 차이점을 부각해보기</li> <li>▪ 우리도시의 좋은 점을 소개하기</li> <li>▪ 우리도시에서 보내는 하루를 구성하고 소개하기</li> </ul> </li> <li>- 패들릿을 활용하여 자유롭게 아이디어를 작성하고 공유하기 <ul style="list-style-type: none"> <li>▪ 모둠별로 게시판을 생성하고 아이디어를 게시하기</li> </ul> </li> </ul>	
	아이디어 수렴 및	<ul style="list-style-type: none"> <li>■ 생성한 아이디어를 수렴하고, 조직화하기</li> <li>- 아이디어 생성 활동을 통해 도출한 아이디어들 중 흥</li> </ul>	

단계	활동 요소	교수 · 학습 활동	자료(☒) 및 유의점(☉)
	조직화	<p>보자료를 만들기에 좋다고 생각하는 아이디어에 공감하기를 누르고, 댓글로 의견을 쓰기(좋은 점, 나쁜 점, 흥미로운 점)</p> <p>- 가장 많은 공감을 얻은 아이디어를 모둠의 최종 아이디어로 정하고, 구현 가능성을 점검하기</p>	
정리	정리 및 차시예고	<p>■ 정리하기</p> <p>- 아이디어 생성과 수렴의 과정에 대해 이야기해보기</p> <p>- 모둠별로 홍보자료의 주제와 도출한 최종 아이디어를 정리하여 패들릿에 공유하기</p> <p>■ 차시예고</p> <p>- 모둠별 환경자료 제작을 위한 스토리보드와 알고리즘을 작성하기</p>	

주 제	환경문제를 알리는 홍보자료(엔트리) 만들기	차 시	3/6	수업 모형	
학습 목표	홍보자료 제작을 위해 도출한 아이디어를 토대로 알고리즘을 설계할 수 있다.				
자 료	교 사	교사용 컴퓨터			
	학 생	크림북, 활동지, 패들렛			

단계	활동 요소	교수 · 학습 활동	자료(☒) 및 유의점(☉)
	동기 유발	<ul style="list-style-type: none"> <li>■ 동기유발 및 이전차시 학습 확인</li> <li>- 이전 차시 제작한 패들릿 보드를 다시 살펴보며 아이디어를 정리하기</li> <li>- 모듈별 도출한 최종 아이디어를 다른 모듈에게 소개하기</li> <li>■ 스토리보드와 알고리즘 설계에 대한 예시를 살펴보기</li> <li>- ‘라면 끓여먹기’와 같은 실생활 속의 현상과 경험을 주제로 스토리보드 구성하기</li> <li>- 스토리보드를 바탕으로 알고리즘과 순서도 형태로 표현한 예시자료를 확인하기</li> </ul>	
	아이디어 구성	<ul style="list-style-type: none"> <li>■ 아이디어를 토대로 홍보자료의 내용을 구성하기</li> <li>- 도출한 최종 아이디어를 바탕으로 스토리보드를 활용하여 내용과 장면을 구성하기 <ul style="list-style-type: none"> <li>▪ 최종 아이디어를 토대로 홍보자료의 내용을 확정하기</li> <li>▪ 내용을 흐름에 따라 장면별로 나누고, 등장하는 인물이거나 물체를 정해 스케치하기</li> <li>▪ 장면별/오브젝트별 대사를 써보기</li> </ul> </li> </ul>	☒ 스토리 보드 활동 지
	논리화	<ul style="list-style-type: none"> <li>■ 알고리즘 작성하기</li> <li>- 스토리보드에 작성한 내용을 구현하기 위하여 어떤 순서대로 알고리즘을 짜면 좋을지 생각하기 <ul style="list-style-type: none"> <li>▪ 구현할 장면을 자연어(문장형태)로 적기</li> <li>▪ 작성한 문장을 스토리보드의 내용을 바탕으로 절차</li> </ul> </li> </ul>	☒ 스토리 보드 활동 지

단계	활동 요소	교수 · 학습 활동	자료(☒) 및 유의점(☉)
		<p>에 따라 잘게 나누기</p> <ul style="list-style-type: none"> <li>▪ 나누어진 문장을 순서도의 형태로 표현하기</li> <li>▪ 장면별로 위 절차를 반복하여 스토리보드 전체를 아우르는 알고리즘을 짜기</li> <li>▪ 작성한 알고리즘이 적절한지 검토하기</li> </ul> <p>- 일부 알고리즘 구성요소를 바탕으로 알고리즘을 발전 시키기(리믹스활동)</p>	<p>☒ 알고리즘 구성요소를 정리한 PPT 자료</p>
정리	정리 및 차시예고	<ul style="list-style-type: none"> <li>■ 정리하기 <ul style="list-style-type: none"> <li>- 알고리즘을 작성하기 위한 과정에서 느낀 점을 이야기해보기</li> <li>- 모듈별로 도출한 알고리즘을 위두랑 플랫폼에 게시하고 다른 모듈의 산출물에 서로 댓글 달기</li> </ul> </li> <li>■ 차시예고 <ul style="list-style-type: none"> <li>- 각 모듈별 알고리즘을 바탕으로 엔트리를 통해 코딩하기</li> </ul> </li> </ul>	

주 제	환경문제를 알리는 홍보자료(엔트리) 만들기	차 시	4~5/6	수업 모형	
학습 목표	설계한 알고리즘을 바탕으로 환경문제를 알리는 홍보자료를 컴퓨팅 산출물의 형태로 제작할 수 있다.				
자 료	교 사	교사용 컴퓨터			
	학 생	크롬북, 활동지, 패들렛			

단계	활동 요소	교수 · 학습 활동	자료(☒) 및 유의점(☕)
	동기 유발	<ul style="list-style-type: none"> <li>■ 동기유발 및 이전차시 학습 확인</li> <li>- 모둠별로 제작하기로 한 홍보자료의 스토리보드를 확인하기</li> <li>- 설계한 스토리보드와 알고리즘을 다시 점검하기</li> <li>- 엔트리로 표현할 수 있는 다양한 형태의 자료를 찾아보기</li> </ul>	
	문제 해결 (프로그래밍)	<ul style="list-style-type: none"> <li>■ 홍보자료 제작에 필요한 다양한 자료를 찾아보기</li> <li>- 스토리보드와 알고리즘 설계 결과를 바탕으로 엔트리 화면 상에 나타내기 위해 필요한 이미지, 오브젝트, 음원 등의 자료를 수집하기</li> </ul>	
		<ul style="list-style-type: none"> <li>■ 엔트리로 코딩하기</li> <li>- 동료프로그래밍 활동을 중심으로 코딩하기 <ul style="list-style-type: none"> <li>▪ 두 학습자 또는 팀을 운전자 역할, 네비게이터 역할로 나누기</li> <li>▪ 운전자 역할의 학생은 키보드와 마우스를 사용하여 타이핑을 하고 코드를 작성/입력 역할을 담당</li> <li>▪ 네비게이터 역할의 학생은 프로젝트의 큰 그림에 집중하고 작성된 코드가 논리적이고 오류가 없는지, 확인하는 역할은 담당</li> </ul> </li> <li>- 4차시와 5차시 진행 시 동료프로그래밍 역할 바꾸기</li> </ul>	
		<ul style="list-style-type: none"> <li>■ 디버깅 및 리믹스 활동하기</li> </ul>	

단계	활동 요소	교수 · 학습 활동	자료(자) 및 유의점(유)
		<ul style="list-style-type: none"> <li>- 엔트리로 코딩하며 작성한 결과값을 반복적으로 확인하고 디버깅 해보기</li> <li>- 온라인 상에서 교사나 다른 학습자들이 만들어 놓은 프로그램의 코드를 열람하고 일부 구성요소를 참고하여 알고리즘을 발전시키기</li> </ul>	
	정리 및 차시예고	<ul style="list-style-type: none"> <li>■ 정리하기               <ul style="list-style-type: none"> <li>- 모듈별로 코딩한 산출물을 엔트리 온라인 클래스에 공유하고 친구들의 작품에 댓글 달기</li> <li>- 스토리보드를 코딩 프로그램을 통해 표현할 때 재미있었던 점/어려웠던 점을 이야기해보기</li> </ul> </li> <li>■ 차시예고               <ul style="list-style-type: none"> <li>- 엔트리로 작성한 모듈별 홍보자료 발표하기</li> <li>- 다른 친구들의 작품을 감상하고 평가하기</li> </ul> </li> </ul>	

주 제	환경문제를 알리는 홍보자료(엔트리) 만들기	차 시	6/6	수업 모형	
학습 목표	학습자가 제작한 홍보자료를 발표 및 공유할 수 있다.				
자 료	교 사	교사용 컴퓨터			
	학 생	크롬북, 활동지, 패들렛			

단계	활동 요소	교수 · 학습 활동	자료(☞) 및 유의점(☞)
	동기 유발	<ul style="list-style-type: none"> <li>■ 동기유발 및 이전차시 학습 확인</li> <li>- 이전 차시에서 만든 코딩 산출물을 확인하기</li> <li>- 각 모듈별로 어떤 홍보자료\를 만들었는지 이야기하 기</li> </ul>	
	발표 및 공유	<ul style="list-style-type: none"> <li>■ 엔트리로 제작한 홍보자료 산출물 점검하기</li> <li>- 스토리보드와 알고리즘을 바탕으로 장면 구성과 코딩 한 것을 대조하고 확인하기</li> </ul>	
		<ul style="list-style-type: none"> <li>■ 최종 산출물을 발표하고 공유하기</li> <li>- 모듈별로 최종 산출물을 시연하고 발표하기(산출물의 주제, 특징, 제작과정 등)</li> <li>- 다른 모듈의 산출물을 보고 잘된 점, 아쉬운 점, 개선 이 필요한 점에 대한 의견을 말하기(온라인 플랫폼에 댓글 달기, 직접 발표하기)</li> </ul>	
	정리 및 마무리	<ul style="list-style-type: none"> <li>■ 정리 및 마무리하기</li> <li>- 엔트리를 활용하여 홍보자료를 제작한 경험을 되돌아 보며 성찰일지를 작성하기</li> </ul>	

Abstract

A Study on Development of an  
Instructional Design Model of  
Integrating Subject Matter and  
Software for Creative Problem  
Solving

Hyunkyung Chee

Department of Education

The Graduate School

Seoul National University

With the coming Fourth Industrial Revolution and the advancement of science and technology, it has become essential for people to acquire new competencies in the educational realm, as it is predicted that new jobs will emerge due to the development of new knowledge and computer algorithms, while existing jobs rapidly disappear as artificial intelligence replaces human labor. This will call for a new workforce that understands new industries and is equipped with creative problem solving skills and both convergence and critical

thinking abilities. Therefore, it is urgently necessary to cultivate talents in these areas. As part of an effort to nurture these next-generation talents, the new software education courses have been included in a regular curriculum for elementary, middle, and high schools since 2015. The goal of the software education program is to foster creative convergence talents by naturally introducing future generations to the daily process of creatively solving complex problems.

Despite its new advantages and possibilities for the future society, the education experts criticized software education for lacking systematic linkage with and relevance to other subjects. In elementary schools, while the software courses are part of the practical subjects, they are structured separately without any systematic linkage with the actual units. On the other hand, in middle and high schools, since the information curriculums are organized and conducted separately, it is challenging to achieve a convergence with other subjects. To overcome these problems, studies were conducted recently to create subject-centered convergence software education programs that cover mathematics, science, art, and other STEAM-related subjects, and these aim to link between software and convergence talent education. In most cases, in the context of computer pedagogy, after the application of software education programs for information and practicum management, the effectiveness of these programs is either verified or further researched with a focus on the teaching/learning model or class model. In addition, several research projects present class models that suggest the procedures and methods of class instructions, such as a course-specific program or teaching plan. The conceptual framework of instructional system design, however, fails to consider various

factors when designing convergence lessons, and it lacks an explanation of the overall process and procedures.

The purpose of this study is thus to develop an instructional model for academic subject–software convergence classes that merge and link the knowledge of existing subjects such as Korean, sciences, and social studies with software lessons in the context of educational engineering. To this end, the research questions address the subject matter and software convergence class design for creative problem–solving skills, and the subsequent development of the teaching design principles, conceptual models, and procedural models. Then, the paper reviews the reaction of the instructors and learners to the class design model that has been developed.

In this study, the research method on design and development of an subject matter and software class is applied, and the procedure is as follows: to begin with, it examines the components and design principles of the model through a review of previous studies. Then, an initial model, including a conceptual model and a procedural model, is developed and validated through an expert validation review, usability evaluation, and field evaluation. Eight instructional design experts participated in two expert validation reviews, and three elementary school teachers participated in usability evaluations. In the field evaluation, one elementary school homeroom teacher and 23 students in the class participated. In the controlled environment, the teacher applied the initial model in the class, and the students responded to the survey questions about the effect of the convergence class on their creative problem–solving skills and academic self–efficacy before and after the lessons. Finally, student interviews confirmed their in–depth opinions.

As a result of this study, a conceptual model, a procedural model,

12 design principles, and 33 detailed guidelines have been developed to guide the subject matter and software convergence class for creative problem solving. The conceptual model consists of five components: subject matter and software convergence curriculum, subject matter and software convergence class activity, subject matter and software convergence learning support, sharing and evaluation, and subject matter and software convergence learning tool. The procedural model included 12 detailed instructional design procedures according to the processes of analysis, design and development, and execution and evaluation. The result from field application showed that classes with this model lead to a positive effect on learners' creative problem solving skills. Lastly, it confirmed that it is an effective model to guide teachers to systematically design the subject matter and software convergence classes for creative problem solving skills.

On the basis of the above research results, the author discussed the subject matter and software convergence class design model, subject matter and software convergence class support, and their theoretical and practical implications. Finally, the author proposed follow-up studies: studies that can check various elements that influence a class designed based on the subject matter and software convergence class design model, studies that can apply various reflective methods, studies that establish teaching/learning strategies and an effective classroom environment to effectively run classes in a poor learning environment with limited infrastructure, studies that systematically yield to the collaboration between schools and institutions, and studies that develop programs to strengthen pre-service teachers' subject matter and software convergence class competency.

keywords : Integrating Subject Matter and Software,  
Computational Thinking, Creative Problem Solving,  
Instructional design model

*Student Number* : 2016-32850