

2015년 교육정책네트워크 교육현장지원연구

SW교육 교수학습 모형 개발 연구

연구책임자: 김진숙 (한국교육학술정보원)

공동연구진: 한선관 (경인교육대학교)

김수환 (충신대학교)

정순원 (한국교육학술정보원)

양재명 (한국교육학술정보원)

장의덕 (한국교육학술정보원)

김정남 (한국교육학술정보원)

연구보조원: 류미영 (경인교육대학교)

이진태 (경인교육대학교)

전수진 (부천부곡초등학교)

김상홍 (풍무초등학교)

본 연구는 한국교육개발원에서 지원한 연구비로 수행한 것으로써, 이 연구에서 제시된 의견은 한국교육학술정보원의 공식 의견이 아니라 연구진의 견해임을 밝힙니다.

연구에 도움을 주신 분들(가나다 순)

SW교육 교수학습 수업자료 개발

김재남(진관중학교)

김형기(인하대학교사범대학부속중학교)

서인순(오산정보고등학교)

신갑천(임진초등학교)

최 만(봉선중학교)

머 리 말

산업사회에서 정보사회로 패러다임이 변화하면서 사회 전반에 많은 변화가 일어나고 있다. 특히 정보가 중요한 가치가 되는 지식정보사회에서 고성능 프로세서, 초고속 네트워크, SW 기술의 융·복합이 강조되는 소프트웨어 중심 사회로 전환되고 있다. 즉, 지식정보사회는 인프라-하드웨어 중심의 1차 IT혁명을 거쳐 창의적 IT인재를 바탕으로 하는 소프트웨어·콘텐츠 중심의 2차 IT혁명 시대로 이동하고 있다. 소프트웨어는 “Software is eating the world”, “Software is everywhere” 라고 표현할 정도로, 자동차산업, 항공산업, 금융산업, 쇼핑산업 등 거의 모든 산업에 미치는 영향이 막대하다.

우리 정부는 이와 같은 사회변화에 대응하고자 2014년 『소프트웨어(SW) 중심사회 실현 전략』을 수립하여 발표하고, 교육부에서는 ‘초·중등 소프트웨어(SW) 교육 활성화 방안’에 이어 『2015년 문·이과 통합형 교육과정』을 통해 초·중·고등학교에 체계적인 소프트웨어(SW) 교육을 도입하였다. 그러나 『2015 문·이과 통합형 교육과정』에 따른 소프트웨어교육이 학교현장에 안정적으로 정착되기 위해서는 교원수급, 교원연수, 교재 개발 등 다양한 영역에서의 준비가 필요하다.

본 연구는 이와 같은 측면에서 일반학교에서 소프트웨어교육을 쉽고 효과적으로 할 수 있도록 교수학습모델을 개발하여 보급함으로써 교사들의 수업을 지원하고자 하였다. 특히, 기존 교수학습모델에 기반을 두어 소프트웨어교육이 추구하는 컴퓨팅사고력을 기르고, 소프트웨어교육의 특징을 잘 살릴 수 있는 교수학습모델을 개발하는데 주력하고자 하였다. 연구 결과가 현장에 도움이 되기를 기대한다. 마지막으로 본 연구를 위해 애써주신 연구진들과 외부 학계 전문가, 학교 현장 전문가들께 감사드린다.

2015년 12월
한국교육학술정보원
원 장 임 승 빈

임승빈

연구 요약

2015년 문·이과 통합형 교육과정에서 적용될 소프트웨어교육의 성공적 현장 적용은 현장 교원의 교육과정, 즉, 교육목표와 방법, 평가, 그리고 운영 편제의 이해가 전제될 때 가능하다. 그동안 소프트웨어교육 운영지침 개발과 교원 연수를 통해 교육과정을 이해하기 위한 기반이 마련되고 있지만, 2016년도부터 확대될 소프트웨어교육 선도학교나 연구학교에서 실제로 적용될 교수학습 방법에 대한 제시는 미흡한 실정이다. 따라서 본 연구는 SW교육의 목표로서 컴퓨팅 사고력과 구성요소, 교육과정 및 교재의 내용 분석에 근거하여 총 5가지의 교수학습 모델을 개발하고 각각의 모델에 적합한 사례를 제시함으로써 소프트웨어교육의 성공적 현장 적용을 지원하는데 그 목적이 있다.

컴퓨팅 사고력의 개념과 구성요소에 대한 여러 논의를 살펴보면 학자에 따라 컴퓨팅 기술을 이용할 수 있는 역량에 초점을 두는 경우, 일의 처리 절차를 공식화하거나 절차적 사고에 초점을 둔 문제해결방법론의 하나로 보는 관점, 기제나 패러다임이라는 인간의 사고에 중점을 둔 경우, 광의적으로 자동화와 정보처리에 대한 연구가 확장되어 가는 과정으로 보는 등으로 다양하다. 본 연구에서는 학교 현장에서 SW교육의 안정적 정착이라는 초점을 두어 ‘컴퓨팅 사고력’이란 학생들의 일상생활에서 발생할 수 있는 문제들을 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력’이라고 정의하였다. 또한, 컴퓨팅사고력의 구성요소에 대해서도 Barr와 Stephenson(2011)과 같은 학자는 12단계로 설명하거나, CSTA(Computer Science Teachers Association, 2011)는 6가지의 특성으로 규정하고 있으나, 구성요소 각각의 요소들이 독립되어 의미를 가지는 경우도 있고, 이것이 융합되어 의미를 가지는 경우도 있다. 그러나 SW교육이 학교에 처음으로 적용되는 단계에서 구성요소가 너무 복잡하게 제시될 경우 가져올 수 있는 혼란이나 교수학습의 어려움 등을 종합적으로 고려하여 본 연구에서는 분해(Decomposition), 패턴인식(Pattern recognition), 추상화(Abstraction), 알고리즘(Algorithms) 등 4가지 요소로 제시하고, 프로그

래밍(Programming)은 선택적으로 포함될 수 있다고 보았다.

이와 같은 이론적 배경 하에 2015 개정 교육과정이 적용되는 2018년 이전까지 SW교육 실시를 위해 마련된 소프트웨어교육 운영지침의 주요내용과 2015 개정 교육과정에서 실과, 정보과의 주요 내용, 성취기준, 시수 기준 등에 대하여 분석하고, 운영지침에 기반을 둔 학생용 교재의 내용을 분석하여 컴퓨팅 사고력 신장을 위한 교수학습 모델과 수업 자료를 제시하였다.

SW교육 교수학습모델은 행동주의, 인지주의, 구성주의적 관점을 고려하여 학습목표 영역인 지식, 기능, 태도 중 기능 영역을 중심으로 수업모델을 개발하되, 기능 이외에 지식 및 태도 영역이 포함될 수 있도록 하였다. 또한 문제해결학습 등 다양한 기존 수업방법에서 빈번하게 다루어지는 모델에 착안하여 새로운 모델을 개발하고, 학습자의 특성, 학교 환경 등을 고려하여 재구성이 가능하도록 하였다. 특히 개발된 수업모델은 컴퓨팅 사고 신장이라는 목표를 달성하기 위해 각 단계별 활동 내에 컴퓨팅 사고 구성요소(분해, 패턴인식, 추상화, 알고리즘, 프로그래밍)를 포함하도록 구성하는 등의 기본방향을 설정하고 총 5개의 모델을 개발하여 제안하였다. 즉, 컴퓨팅 사고의 구성요소가 모든 학습의 목표이자 기초가 되어야 한다는 점에서 CT요소중심모델(DPAA(P) 모델), 시연중심모델(DMM 모델), 재구성중심모델(UMC 모델), 개발중심모델(DDD 모델), 디자인중심모델(NDIS 모델) 등 5가지로 특성화하여 개발하였다. 각각의 모델에 대해서는 모델의 개요, 학습 목표, 단계별 세부 내용 등을 설명하여 각 모델이 갖는 특징과 방법에 대하여 다루었고, 실제 수업에서 어떻게 적용될 수 있는지에 대해서는 각 모델마다 학교급별 수업 자료를 제시하였다. 특히 각 모델들이 단순히 연구에서 그치는 것이 아니라 차후 교사 연수과정이나 혹은 교사 개인적 교수법 연구 시 활용될 수 있도록 가급적 상세히 내용을 설명하고자 하였다.

컴퓨팅 사고력의 구성요소를 단순화하여 교수학습에 이용할 수 있는 모델을 제안하고, 향후 3년 동안 실제 수업에 사용될 사례를 제시한 본 연구는 앞으로 실천적 수업을 이끌어 갈 교사들에게 수업 방향을 제시하고 있다는 점에서 의의를 찾아볼 수 있다. 본 연구에서 제안된 5가지 SW교육 교수학습 모델은 다음과 같다.

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

차 례

요 약	i
I. 서론	1
1. 연구의 목적 및 필요성	1
2. 연구의 내용 및 범위	3
3. 연구 방법	4
4. 용어의 정의	5
5. 연구의 제한점	6
II. 이론적 배경	7
1. 컴퓨팅 사고력	7
2. 국내외 SW교육 교육과정	13
3. 교수학습 모델 선행연구 분석	34
III. 컴퓨팅 사고력 신장을 위한 교수학습 모델 개발	53
1. 교수학습 모델의 개발 준거	53
2. 5가지 CT 신장 교수학습 모델 개발	57
IV. 결론 및 제언	79
1. 요약 및 결론	79
2. 시사점 및 제언	81
참고문헌	82
영문초록	87
부 록	89
1. CT 교수학습모델을 적용한 수업 자료	90
2. SW교육 운영지침과 2015 개정 교육과정 비교	155

표 차 례

<표 II-1> 컴퓨팅 사고력에 대한 다양한 논의	8
<표 II-2> 컴퓨팅 사고력의 구성요소	12
<표 II-3> CT 구성요소의 재정의	12
<표 II-4> 해외 SW교육 동향	13
<표 II-5> 소프트웨어교육 인재상	14
<표 II-6> 2015 개정 교육과정 주요 개편 사항	14
<표 II-7> 소프트웨어교육 학교급별 교육 목표	16
<표 II-8> 소프트웨어교육 초등학교 내용 요소와 성취기준	17
<표 II-9> 소프트웨어교육 중학교 내용 요소와 성취기준	18
<표 II-10> 소프트웨어교육 고등학교 내용 요소와 성취기준	20
<표 II-11> 2015 개정 교육과정에서의 소프트웨어교육	22
<표 II-12> 초등학교 실과 기술시스템 영역 내용 요소와 성취기준	23
<표 II-13> 중·고등학교 정보과 영역별 내용 요소	24
<표 II-14> SW교육 교재의 모듈별 시수	27
<표 II-15> 초등학교 모듈 구성안	27
<표 II-16> 중학교 모듈 구성안	28
<표 II-17> 지식·활동 모듈의 구성 요소	29
<표 II-18> The Big 6 Skills	37
<표 II-19> 디자인 기반 학습의 단계	39
<표 II-20> UMC 모델	42
<표 II-21> DDC 모델	42
<표 II-22> DCC 모델	43
<표 II-23> 창의 컴퓨팅 평가방법	43
<표 II-24> CT 역량 평가 루브릭	44
<표 II-25> 기존 수업과 플립 러닝 적용 수업 구조 및 활동내용 비교	47
<표 II-26> ADDIE 기반 플립 러닝 교수학습 설계 모델	48

<표 III-1> 전통적인 교수방법과 CT 신장 교수학습모델 비교	58
<표 III-2> DMM 모델 개요	60
<표 III-3> DMM 모델의 교수학습 절차	60
<표 III-4> 발견학습법 수업 절차	62
<표 III-5> UMC 모델 개요	63
<표 III-6> UMC 모델의 교수학습 절차	64
<표 III-7> 탐구학습법 수업 절차	66
<표 III-8> DDD 모델 개요	66
<표 III-9> DDD 모델의 교수학습 절차	68
<표 III-10> 프로젝트학습법 수업 절차	70
<표 III-11> NDIS 모델 개요	71
<표 III-12> NDIS 모델의 교수학습 절차	72
<표 III-13> 문제해결학습법 절차	75
<표 III-14> DPAA(P) 모델 개요	75
<표 III-15> DPAA(P) 모델의 교수학습 절차	77
<표 III-16> 5가지 SW교육 교수학습 모델	78

그림 차례

[그림 II-1] 지식 모듈 예시	30
[그림 II-2] 기능 모듈 예시	30
[그림 II-3] 활동 모듈 예시	31
[그림 II-4] 활동 모듈 개요 페이지 예시	32
[그림 II-5] 문제해결 방법찾기 예시	33
[그림 II-6] 알고리즘 만들기 예시	33
[그림 II-7] 프로그래밍하기 예시(스크래치/엔트리틀 같이 제공)	34
[그림 II-8] 문제중심학습의 일반적 절차	36
[그림 II-9] 디자인 사고의 개념	38
[그림 II-10] 디자인 사고 교수학습 절차	40
[그림 II-11] 창의 컴퓨팅 디자인 기반 학습 절차	41
[그림 II-12] 플립 러닝의 개념도	46
[그림 II-13] 스마트교육 기반 플립러닝 수업모델	51
[그림 II-14] SW교육을 위한 플립러닝의 수업 흐름	52
[그림 III-1] CT 신장을 위한 교수학습 모델 개요도	59
[그림 III-2] MIT의 Creative Computing	63
[그림 III-3] 소프트웨어공학설계의 폭포수 모델	67
[그림 III-4] DBL 학습과정	71
[그림 III-5] KS3와 구글에서 제시한 CT의 구성요소	76

I. 서론

1. 연구의 목적 및 필요성

현 사회는 지식정보화 사회를 뛰어넘어 소프트웨어(SW)가 국가의 경쟁력을 결정하는 'SW중심사회'로 진입하고 있다. 무인자동차 기술의 선두주자는 완성차 업체가 아닌 정보통신기술 기업인 구글이다. 소프트웨어의 발달로 산업 간 경계를 허물고 사회의 모든 영역에서 구글 등과 같은 SW 기업들의 영향력이 점차 커지면서, 이제 소프트웨어 기술이 국가 경쟁력의 중심이 되는 'SW중심사회'가 도래한 것이다(SPRI, 2014).

SW중심사회를 이끌어 갈 미래 인재 양성을 위해 국내외의 많은 나라에서는 소프트웨어교육(이하 SW교육이라 한다)을 앞 다투어 추진하고 있다. SW교육은 기존 정보통신기술 교육에서 수행하였던 ICT 소양 및 활용 교육의 관점을 확장하고, 학습자들이 미래사회에서 살아가는데 필요한 컴퓨팅 사고력(Computational thinking, 이하 CT라 한다)을 기반으로 문제를 해결하는 역량을 기르는데 목적을 두고 있다. 영국은 2014년부터 초·중등 교육과정에 '컴퓨팅' 교육을 필수화하였고, 미국은 K-12 컴퓨터 과학 표준에 컴퓨팅 사고력을 명시하였다. 프랑스는 SW교육을 중학교 정규 과목으로 개편하였으며, 인도는 '프로그래밍' 과목을 초중등 필수 과목으로 지정하였다. 그리고 에스토니아는 2015년부터, 핀란드는 2016년부터 모든 초등학생을 대상으로 SW교육을 실시하는 등 많은 나라들이 초등학교에서부터 정보 교육을 실시하고 있으며, 기존의 ICT 활용 중심에서 알고리즘을 중심으로 한 SW교육으로 전환하고 있다(교육부·KERIS, 2015).

우리나라에서도 세계적으로 미래 경쟁력 확보를 위한 2015 개정 교육과정을 마련하고 SW교육을 강화하기 위해 준비하고 있다. 교육부는 「소프트웨어 교육 운영지침」 및 「2015 문이과 통합형 교육과정 총론」을 발표하면서(교육부·KERIS, 2015; 교육부, 2015), 초등학교에서는 실과 교과에서 SW 기초 소양을

중심으로 문제해결 과정, 알고리즘 및 프로그래밍 체험, 저작권 보호 등 정보 윤리 의식 함양 등을 포함하는 내용으로 17시간 교육을 실시하기로 하였다. 또한 중학교에서는 전문적인 SW교육을 위해 '정보' 과목을 의무화하여 컴퓨팅 사고 기반 문제해결 교육과 알고리즘 및 프로그래밍 개발 등을 주요 내용으로 총 34시간 교육하도록 하였으며, 고등학교에서는 심화선택이던 '정보' 과목을 일반 선택으로 전환하여 다양한 분야와 융합한 알고리즘의 설계 및 프로그램 개발 중심의 교육을 실시하는 한편, 기초과목과 실무과목으로 개편하여 NCS 기반의 교육과정을 구성하기로 하였다.

이와 같이 2015 개정 교육과정이 적용되는 2018년이 2년 남짓으로 다가오고, 2016년부터 SW교육 선도학교가 900여개로 확대되는 시점에서 SW교육이 학교 현장에 성공적으로 정착되기 위해서는 교사가 SW교육과정에 대한 이해를 기반으로, 수업을 기획, 설계하고 경험하는 것이 중요하다. 이외에도 교육환경의 점검 및 개선, 교원의 확보 및 연수 등 산적한 과제들이 많으나, 무엇보다 당장 급선무인 것은 SW교육을 수업시간에 어떻게 가르칠 것인지에 대한 교수학습방법을 이해하고 숙지하는 것이다.

컴퓨팅 사고력을 중점에 둔 교수학습 설계 및 모델 개발 관련 연구는 다수 이루어져 왔으나(김수환, 한선관, 2012; 최형신, 2014; 김수환, 2015; 전용주, 김태영, 2015; 한선관, 2015), 2015 교육과정 개정 방향에 기반을 둔 교수학습 모델, 설계 전략 등에 대한 연구는 부족한 상황이다. 따라서 본 연구는 2015 교육과정에서 명시된 SW교육 목표와 방향에 맞춘 교수학습 모델과 실제 사례를 제시함으로써 SW교육의 성공적 현장 안착을 지원하는데 그 목적이 있다.

2. 연구의 내용 및 범위

본 연구에서 다루고 있는 연구 내용과 범위는 다음과 같다.

- 컴퓨팅 사고력 및 SW교육 교육과정 분석
 - SW교육의 목표로서 컴퓨팅 사고력의 개념 및 구성요소 분석
 - 소프트웨어교육 운영지침 및 2015 개정 실과/정보과 교육과정 분석
 - 소프트웨어교육 운영지침에 따른 초·중 학생용 교재 개발 방향 및 내용 분석
- SW교육의 학습요소, 학습도구, 학습환경 등을 고려하여 CT를 증진할 수 있는 교수학습 모델 개발
 - CT의 핵심 구성 요소를 고려한 교수학습 모델 개발
 - 언플러그드, EPL(Education Programming Language), 피지컬 컴퓨팅 등 SW교육의 다양한 학습도구와 학교의 교실환경을 고려하여 학교 현장에 적합한 모델 개발
- 교수학습 모델의 프로토타입에 대한 다양한 교수학습 사례 개발
 - 개발된 교수학습 모델을 적용한 SW교육 교수학습 예시 개발
 - 수업 적용 가능성을 고려한 학교급별 사례 제시
- 모델개발에 따른 시사점 도출 및 제언
 - 컴퓨팅 사고력 증진 교수학습 모델 학교 현장 적용방안
 - 교수학습 모델 적용 유의점

3. 연구 방법

가. 문헌 분석

본 연구에서는 아래와 같은 내용을 중심으로 문헌분석을 실시하였다.

- 국내 및 해외 SW교육 동향 조사 및 분석
- SW교육 교육과정 분석
 - 정부부처 보도 자료 및 연구학교/선도학교 성과 보고
 - SW교육 운영지침 및 2015 개정 교육과정(실과/정보과)
- 컴퓨팅 사고력
 - 컴퓨팅 사고력에 대한 국내외 선행연구 분석
 - NRC(National Research Council), CSTA(Computer Science Teachers Association), 영국 KS3 CS(Key Stage 3, Computer Science) 교육과정, Google's CT 중심으로 조사 및 분석 수행
- SW교육 관련 교수학습 모델
 - 국내외 교수학습 모델 선행연구 분석
 - SW교육과 관련한 Creative Computing(MIT 미디어랩, 하버드대학교 Brennen을 중심으로)과 Design Thinking(스탠포드대학교 D-school을 중심으로) 조사 및 분석

나. 전문가 회의

본 연구에서는 내용 타당도를 확보하고, 학교 현장에의 적용 가능성을 높이기 위해 전문가 회의를 수행하였다. 본 연구의 전문가 회의에는 SW교육 관련

학계 전문가, 2016년 학교에 배포되는 SW교육 교재 집필진, 연구학교/선도학교 교원이 함께 참여하였다.

- 교수학습 모델 개발을 위한 전문가 회의(학계, 현장교사 참여)
 - 컴퓨팅 사고력의 구성요소 분석
 - 교수학습 모델 개발에 대한 이론적 근거 검토
 - SW교육 수업에 적합한 교수학습 모델 및 방법
 - SW교육 운영지침 및 2015 개정 교육과정 분석 결과에 따른 모델 반영 방법
 - CT 신장을 위한 교수학습 모델의 설계
 - 개발 모델에 대한 현장 적합 가능성 검토
 - 사례 개발 및 상호 검토(다각 검증)

4. 용어의 정의

- 컴퓨팅 사고력(Computational Thinking) : 학생들의 일상생활에서 발생할 수 있는 문제들을 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력
- 컴퓨팅 사고력의 구성요소 : 컴퓨팅 사고력을 구성하는 분해, 패턴 인식, 추상화, 알고리즘, 프로그래밍의 5가지 요소¹⁾
 - 분해(Decomposition) : (자료와 과정을 포함하여) 어떠한 문제를 다룰 수 있는 작은 것으로 나누는 것
 - 패턴 인식(Pattern recognition) : 어떤 문제나 현상에서 반복되는 성질의 것 또는 그러한 경향이나 규칙을 찾는 것, 패턴 찾기(finding patterns)
 - 추상화(Abstraction) : 패턴을 만드는 일반 원칙을 정하는 것 또는 모델링이나 시뮬레이션을 통해 자료를 표현하는 과정 전체

1) CSTA(2011), 영국 KS3 Computer Science, Google's CT lesson plan을 참조하여 제정의.

- 알고리즘(Algorithm) : 문제를 풀기 위한 단계적인 절차 또는 문제를 해결하기 위한 일련의 단계를 절차적으로 표현하는 과정 전체
- 프로그래밍(Programming) : 문제를 풀기 위해 컴퓨터가 해야 할 일을 컴퓨터가 이해할 수 있는 언어(Programming language 또는 code)로 만들어보는 것 또는 문제를 해결하기 위한 프로그래밍과 이를 실행하는 과정 전체

○ 교수학습 모델 : 수업의 실재를 기술하기 위하여 수업의 주요 특징을 요약해 놓은 설계도 또는 계획으로 수업현상을 기술하고 설명할 수 있으며, 나아가 예언할 수 있도록 수업의 주요 특징을 간추려 체계화시켜 놓은 형태. 즉, 복잡한 수업 현상이나 수업 사태를 그 특징을 중심으로 단순화시킨 형태이며 수업 현상을 구성하는 변인들 간의 관계를 단순화시킨 형태를 의미

○ 교수학습 전략 : 수업 목표를 도달하기 위해 사용하는 모델의 복합적 구성 또는 모델 내의 단계 내에서 이루어지는 활동들을 효과적으로 지원하기 위한 교수학습 방법 및 전략을 의미

5. 연구의 제한점

본 연구는 2015 개정 교육과정이 적용되는 2018년 이전까지 시행되는 소프트웨어교육 운영지침(2015, 교육부)과 이를 기반으로 개발된 학생용 교재의 내용을 참고하여 교수학습모델을 개발하였다. 따라서 2015 교육과정에 바로 적용하고 일반화시키는데 한계를 가진다. 또한 본 연구에서 제안된 수업 사례는 교수학습 모델별 절차에 충실한 아이디어 수준의 사례를 제시하였다. 따라서 모든 수업 환경에서 일반화하는데 한계를 가진다. 하나의 교수학습 모델이 다양한 학습환경에서 적용될 수 있는 사례 확보와 이를 적용한 학습 효과 검증 등의 연구가 후속적으로 이루어질 필요가 있다.

II. 이론적 배경

1. 컴퓨팅 사고력

SW교육의 중요성과 필요성은 충분히 인식되고 있지만, SW교육이 실효를 거두기 위해서는 교육과정의 목표와 내용이 구체화되고, 교육과정 운영이 융통성이 있어야 한다는 지적이 있어왔다(김영수, 홍경선, 2000; 박종호, 박수중, 김종훈, 2009; 서수영, 김윤호, 2003; 유인환, 구덕희, 2004; 장용준, 한선관, 2007; 최숙영, 2011). 최근 SW교육의 목표에 대해서는 대체적으로 학생들의 컴퓨팅 사고력 증진에 중점을 두어야 한다는 것에 일관성 있는 보고가 이어지고 있다(이영준, 이은경, 2008; 김수환, 한선관, 2012, 성정숙, 김현철, 2015). 실제 2015 교육과정의 '정보' 교과 교육과정에서도 컴퓨팅 사고력을 목표로 두고 있다(교육부·KERIS, 2015). 이하에서는 컴퓨팅 사고력(이하 CT)의 개념과 구성요소에 대해 살펴보고자 한다.

가. 컴퓨팅 사고력의 정의

SW교육의 목표로 CT의 중요성은 여러 학자들에 의해 주장되고 있다(교육부·KERIS, 2015; 유중현, 김종혜, 2008; 이영준, 이은경, 2008; 정인기, 2014; 최숙영, 2011). CT에 대한 논의는 CT가 읽기, 쓰기, 셈하기와 마찬가지로 21세기를 살아가는 모든 사람들이 갖추어야 할 기본 사고 능력이라고 주장한 Wing(2006)의 논의에 의해 촉발되었다. CT가 하나의 사고 능력이라는 점에 대해 Bundy(2007)는 21세기를 이해하기 위한 인간 인지적 능력의 확장이라고 보는 한편, Kramer(2007)는 사고 능력이라기보다 '사고의 패러다임'적 측면이 강하다고 보았다. 이와 유사한 측면에서 CT를 지적 활동을 위한 기호 체계로 보거나, 추상화 도구로써 인간의 근본적인 정신적 능력이자 사고의 기제

(mechanism)임을 강조한 견해 등은 CT에 대한 논의를 언어적 측면으로까지 발전시킨 바 있다(NRC, 2010). 이외에도 일을 처리하는 방법의 공식화라든가, 문제 해결을 목적으로 무형의 추상적 개념을 다루고 조작하는 것이라는 문제해결 능력이라는 측면에서의 논의가 있었다(NRC, 2010). 비교적 최근 미국에서는 CT는 K-12 컴퓨터 과학 표준 전체에 스며들어 있는 추상화, 자동화, 분석에 중점을 둔 컴퓨터 과학의 핵심 원리이자, 컴퓨터 과학과 모든 학문 분야를 연관시킬 수 있는 컴퓨터로 해결할 수 있는 문제들의 해법을 분석하고 발전시키는 명확한 수단을 제공하는 문제해결방법론의 하나라고 정의한 바 있다(CSTA, 2011).

위와 같은 CT에 대한 논의와 함께, 컴퓨팅이 인공적인 정보처리과정 뿐만 아니라 자연적 정보처리과정까지를 포함하는 학문이라는 컴퓨터 과학 자체에 관한 관점의 변화라는 Denning(2007)의 주장까지를 복합적으로 생각해보면 ‘SW를 어떻게 활용할 것인가’ 하는 도구 활용 능력에 대한 관심에서, ‘SW는 어떻게 작동하고 구현되며, 이러한 원리를 실제 문제의 해결에 어떻게 적용할 수 있는가’ 하는 보다 근본적인 사고력에 대한 차원으로 SW교육의 관심이 이동하고 있음을 알 수 있다.

다양한 학자들이 주장하고 있는 컴퓨팅 사고력의 정의를 정리해 보면 <표 II-1>과 같다.

<표 II-1> 컴퓨팅 사고력에 대한 다양한 논의

학자	컴퓨팅 사고력에 대한 주장
Wing, J. (2006, 2008)	<ul style="list-style-type: none"> · 문제 해결, 시스템 설계 및 인간 행동의 이해를 컴퓨터 공학의 기본 개념을 통해 접근하는 방법(컴퓨터가 아닌 인간의 사고 방법) · 추상화 및 자동화 등 컴퓨터를 활용한 문제해결 능력

〈표 II-1〉 컴퓨팅 사고력에 대한 다양한 논의(앞 장에서 계속)

학자	컴퓨팅 사고력에 대한 주장
Wing, J. (2006, 2008)	<ul style="list-style-type: none"> · CT는 분석적 사고의 하나로, 문제 해결을 위한 일반적인 접근에서는 수학적 사고를, 실제 세계의 제한 안에서 크고 복잡한 시스템 설계와 평가를 하는 문제들에는 공학적 사고를, 인간의 행동·감정·지능·연산 능력을 이해하기 위한 문제의 접근에는 과학적 사고를 함께 사용
Moursund, D. (NRC, 2010)	<ul style="list-style-type: none"> · Papert의 절차적 사고로부터 발전된 개념(Papert, 1981: NRC, 2010에서 재인용)이며, 절차적 사고는 개발(developing), 표현(representing), 시험(testing), 오류 수정(debugging) 순서로 구성 · 효과적인 절차 : 컴퓨터 및 자동화기기를 이용한 과업 수행과 컴퓨터가 이해할 수 있는 세부적인 단계적 명령의 집합
Lee, P. (NRC, 2010)	<ul style="list-style-type: none"> · 인간 지능을 확장하여 실제 응용이 가능한 지능(≠ 인공지능)의 기제(mechanism)에 관한 연구 · 업무 자동화를 가능하게 하고, 복잡함을 관리하는 추상화 도구로서의 인간의 근본적인 정신적 능력 · (McGettrick, A.) 사고 과정에 기술 장비를 다루는 실제 능력과 역량을 포함하는 개념
Wulf, B. (NRC, 2010)	<ul style="list-style-type: none"> · 과학이 물리적 대상(physical objects)에 관한 영역인 것과 대조적으로 어떠한 과정을 가능하게 하는 추상적인 현상과 절차에 관한 것 · (Denning, P.) 컴퓨터 과학은 정보처리를 위한 학문 분야, CT는 컴퓨터 과학의 하위 영역
Abrahamson, D. (NRC, 2010)	<ul style="list-style-type: none"> · 형식지(explicit knowledge)로 표현하거나 암묵지(tacit knowledge)를 객관화하고, 구체적인 전산 용어로 나타내며, 인지적 활동으로부터 발생한 산물을 처리하기 위한 컴퓨팅 관련 기호 체계(semiotic systems)의 사용
Sussman, G. (NRC, 2010)	<ul style="list-style-type: none"> · 일을 처리하는 정확한 방법을 공식화(formulating)하는 기법 · 제시된 과업을 효율적으로 수행하기 위한 엄정한 분석과 수순
Wing, J. & Sussman, G. (NRC, 2010)	<ul style="list-style-type: none"> · 자연과학과 공학을 잇는 메타과학(여러 학문 분야에 걸쳐 적용되는 사고의 연구 방법)으로 과학적 관찰의 응용, 물리적 현상의 연구에서 전이가 발생하는 추론의 핵심 요소

〈표 II-1〉 컴퓨팅 사고력에 대한 다양한 논의(앞 장에서 계속)

학자	CT의 정의
Fox, E. (NRC, 2010)	<ul style="list-style-type: none"> · CT의 핵심은 문제 해결을 목적으로 무형의 추상적 개념을 다루고 조작하는 것(사람이 세상을 접하고, 절차를 고려하고, 디지털 표현을 처리할 때 행하는 것이며, 모든 사람들의 일상에 이미 어느 정도로 관여되어 있는 것) · (Blake, B.) CT는 표현(representations), 시각화(visualizations), 모델링(modeling) 또는 메타 모델링(meta-modeling)을 포함
Constable, R. (NRC, 2010)	<ul style="list-style-type: none"> · 기술과 사고 과정의 유한한 집합이라는 한정적 정의가 아닌, 인지적 과정의 자동화와 정보처리에 대한 연구 정도로 개술되는 모든 묘사를 조합하여 역동적인 기술의 속성과 인간의 학습을 반영한 제한을 두지 않고 발전하고 있는 개념의 리스트
이영준(2008)	<ul style="list-style-type: none"> · 추상적 사고의 자동화 과정인 컴퓨팅 과정에서 문제해결을 위해 적절한 추상적 개념을 선택하고 이를 자동화하기 위해 적절한 컴퓨팅 장치를 선택하기 위한 사고 능력
최숙영(2011)	<ul style="list-style-type: none"> · 창조적인 다중의 추상화 차원의 사고 · 문제해결 과정을 컴퓨팅 시스템에서 처리될 수 있는 형태로 알고리즘화하는 논리적 사고 · 문제해결을 위해 컴퓨팅의 기본 개념을 사용하고 컴퓨터 등의 기기를 다룰 수 있는 인지를 포함하는 기술
CSTA(2011)	<ul style="list-style-type: none"> · 컴퓨터 과학과 모든 학문 분야를 연관시킬 수 있는 컴퓨터로 해결할 수 있는 문제들의 해법을 분석하고 발전시키는 분명한 수단을 제공하는 문제해결방법론의 하나 · K-12 컴퓨터 과학 표준 전체에 스며들어 있는 추상화, 자동화, 분석에 주안점을 둔 컴퓨터 과학의 핵심 원리
교육부 · KERIS (2015)	<ul style="list-style-type: none"> · 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력

지금까지 살펴본 여러 논의는 학자에 따라 컴퓨팅 기술을 이용할 수 있는 역량에 초점을 두는 경우, 일의 처리 절차를 공식화하거나 절차적 사고에 초점을 둔 문제해결방법론의 하나로 보는 관점, 기제나 패러다임이라는 인간의 사고에 중점을 둔 경우, 광의적으로 자동화와 정보처리에 대한 연구가 확장되어 가는 과정으로 보는 등으로 다양하나, 본 연구에서는 학교 현장에서 SW교육의 안정적인 정착이라는 초점을 두어 '컴퓨팅 사고력'이란 학생들의 일상생활에서 발생할 수 있는 문제들을 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력'이라고 정의하고자 한다.

나. 컴퓨팅 사고력의 구성 요소

Wing(2006, 2008)은 CT의 핵심은 추상화와 자동화라고 밝힌 바 있으며, Barr와 Stephenson(2011)은 자료수집, 자료분석, 자료제시, 문제분해, 추상화, 제어구조, 알고리즘 및 절차, 분석과 모델 타당화, 자동화, 병렬화, 시험 및 검증, 시뮬레이션의 12가지 단계로 세분화하여 제시하였다. 한편 NRC(2010) 워크숍에서 David Moursund는 개발(developing), 표현(representing), 시험(testing), 오류 수정(debugging)의 순서로 구성된 절차적 사고임을 강조한 바 있으며, Blake는 CT에는 표현(representing), 시각화(visualization), 모델링(modeling) 또는 메타 모델링을 포함한다는 의견을 피력하기도 하였다.

CSTA(2011)는 이러한 논의를 토대로 CT를 6가지 특징을 지니는 하나의 문제해결 과정이라는 견해를 밝혔으며, 문제해결을 위한 컴퓨팅 사고력의 구성요소를 제시한 바 있다. 이러한 CSTA의 견해는 우리나라의 소프트웨어교육 운영지침에도 반영되어 있다. 교육부에서 제시하고 있는 CT의 구성요소를 제시하면 <표 II-2>와 같다.

〈표 II-2〉 컴퓨팅 사고력의 구성요소(교육부-KERIS, 2015)

구성요소		정의
자료 수집		· 문제해결에 필요한 자료를 모으기
자료 분석		· 자료의 이해, 패턴 찾기, 결론을 도출하기
구조화		· 문제를 그래프, 차트, 그림 등으로 시각화하기
추상화	분해	· 문제를 관리 가능한 수준의 작은 문제로 나누기
	모델링	· 문제 해결을 위한 핵심요소를 추출하고, 모델 만들기
	알고리즘	· 문제를 해결하기 위한 일련의 단계를 알고리즘으로 표현하기 (절차적 표현)
자동화	코딩	· 프로그래밍 언어를 이용해, 문제해결과정을 자동화하기
	시뮬레이션	· 프로그램(소프트웨어)을 실행하기
일반화		· 문제해결과정을 다른 문제에 적용하기

이와 같은 연구에 기반을 두고 볼 때, CT의 구성요소는 대동소이하며 세분화의 정도에 있어 차이가 있는 것으로 생각된다. 세분화된 정도에 따라 너무 잘게 쪼개져있는 경우 실제 교수학습 활동이나, 교과목표를 달성하기에 곤란할 수 있다. 또한, 실제 학습활동은 하나의 요소만을 가지고도 할 수도 있고, 때론 여러 요소가 융·복합되어 이루어지기 때문에 이를 너무 세부적으로 구분하기 어려운 측면도 있다. 이를 위해 본 연구에서는 〈표 II-3〉과 같이 CT의 구성요소를 분해(D), 패턴인식(P), 추상화(A), 알고리즘(A) 등 4가지 요소로 제시하고, 프로그래밍(P)이 포함될 수 있다고 보았다.

〈표 II-3〉 CT 구성요소의 재정의

구분	KS3 용어 ²⁾	Google 용어 ³⁾
분해(D)	Decomposition	Decomposition
패턴 인식(P)	Pattern recognition	Pattern recognition
추상화(A)	Abstraction	Abstraction
알고리즘(A)	Algorithms	Algorithm design
프로그래밍(P)		

2) 영국 key stage 3 Computer science. CT와 Programming을 구분하여 다른 영역으로 제시함.

3) Google for educators. <https://goo.gl/iusIli>

2. 국내외 SW교육 교육과정

가. 해외 SW교육 정책 동향

최근 국내외 SW교육의 방향은 CT 함양을 목표로 정보 교육을 필수화하거나, ICT 활용 중심에서 알고리즘을 중심으로 한 SW교육으로 전환하고 있다(교육부-KERIS, 2015). 해외 주요 국가의 SW교육 목표와 교육과정 동향을 살펴보면 <표 II-4>와 같다.

<표 II-4> 해외 SW교육 동향(교육부-KERIS, 2015; 성정숙, 김현철, 2015)

국가	주요 내용
미국	<ul style="list-style-type: none"> · K-12 컴퓨터 과학 표준(K-12 Computer Science Standards) 발표('11년) · 아칸소주는 고등학교 정규과목에 coding 수업 포함 의무 법제화 · 워싱턴, 텍사스, 켄터키 등은 고등학교 제2외국어 대신 coding 선택
영국	<ul style="list-style-type: none"> · '14.2월 '코딩 교육의 해' 선포, '14.9월부터 초·중등 교육과정에 SW교육 필수화 · 만 5세부터 알고리즘과 디지털 콘텐츠 제작 및 활용 기술 습득 · 1,500여개 학교에서 방과 후 코딩 프로그램 'Code Club' 운영
프랑스	<ul style="list-style-type: none"> · 2016년 9월 신학기부터 SW를 중학교 정규 과목화(외국어 수업 축소) · 이동통신사 프리모바일, 코딩을 무료로 가르치는 에콜 24학교 운영
이스라엘	<ul style="list-style-type: none"> · '94년부터 소프트웨어 과목을 정규과목에 포함 · 고등학교 이과생들을 대상으로 3년간 450시간을 SW수업에 할애
중국	<ul style="list-style-type: none"> · 중·고등학교 정규 교과과정에 SW 필수 교육(초등학교는 선택) · 전국 37개 NPSS(National Pilot School of Software) 운영, 약 6.4만명 재학 중
기타	<ul style="list-style-type: none"> · (일본) '정보' 과목을 고등학교에서 필수과목으로 지정('09년) · (인도) '프로그래밍'과목을 초·중등에서 필수과목으로 지정('13년) · (에스토니아) 2015년부터 모든 초·중학생이 SW교육을 받게 됨 · (핀란드) 2016년 9월부터 모든 초·중학생이 SW교육을 받게 됨

나. 우리나라의 SW교육 교육과정 분석

세계적으로 SW교육을 강화하고 있는 가운데 우리나라는 CT를 기반으로 문제해결역량을 함양하는 SW교육을 강화하기 위해 소프트웨어교육 운영지침과 2015 개정교육과정 실과/정보과 각론이 발표되었다. 소프트웨어교육 인재상 및 교육과정 주요 개편사항은 <표 II-5>, <표 II-6>과 같다.

<표 II-5> 소프트웨어교육 인재상(교육부-KERIS, 2015)

컴퓨팅 사고력을 가진 창의-융합 인재 양성		
↑	↑	↑
초등학교(체험, 활동)	중학교(개념 이해)	고등학교(개발, 융합)
건전한 정보윤리의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해	간단한 알고리즘을 설계하고 프로그램을 개발하여 창의적으로 문제를 해결	효율적인 알고리즘을 설계하고 다양한 분야와 융합하여 문제를 해결

<표 II-6> 2015 개정 교육과정 주요 개편 사항(교육부 2015)

구분	현행	개편안	주요 개편 방향
초등학교 (‘19년~)	실과 內 ICT 단원 (12시간)	실과內 SW교육 실시 (17시간)	- 문제해결과정, 알고리즘, 프로그래밍 체험 - 저작권 보호 등 정보윤리의식 함양
중학교 (‘18년~)	‘정보’과목 (68시간, 선택)	‘정보’과목 (34시간, 필수)	- 컴퓨팅 사고력 기반 문제해결 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

1) 소프트웨어교육 운영지침

2009 개정 교육과정에서 이루어지는 SW교육은 초등학교 5학년 실과 과목의 한 단원, 중학교는 선택 과목(정보)으로서 일부 학교에서 선택적으로 교육하고 있다. 기존의 정보 교육과정의 교육 내용은 정보통신기술을 활용하는데 초점이 맞추어져 있으나, 소프트웨어교육 운영지침의 교육 내용은 정보 윤리를 강화하고 알고리즘과 프로그래밍을 활용한 문제해결력 신장에 중점을 두고 있다고 밝히고 있다(교육부·KERIS, 2015).

소프트웨어교육 운영지침에는 새 교육과정 적용 전까지 연구학교와 선도학교에서 SW 중심의 실과/정보과 교육을 실시하기 위한 목표, 내용, 방법, 평가 등이 담겨있다. 소프트웨어교육 운영지침에서는 우선 CT를 가진 창의·융합인재의 양성, 다시 말해서 단순한 정보통신기술 활용 교육에서 벗어나 소프트웨어로 새로운 사회의 가치를 창출할 수 있는 컴퓨팅 사고력을 지닌 인재 양성을 목적으로 제시하였으며, 이를 위해 학교급간 소프트웨어교육의 연계성을 유지하고, 현재의 교육과정에서 최대한 융통성 있게 소프트웨어 중심 교육을 운영할 수 있는 방안을 제시하여 새로운 운영지침의 적용에 따른 혼란을 줄이는 데 노력을 기울였다고 밝히고 있다.

소프트웨어교육의 내용은 크게 생활과 소프트웨어, 알고리즘과 프로그래밍, 컴퓨팅과 문제해결 3가지 영역으로 제시되어 있으며, 이 가운데 컴퓨팅과 문제해결 영역은 중학교와 고등학교에서만 다루도록 되어 있다. 그리고 학교급 각각의 성취기준 외에도 학교급 간의 성취기준 연계도를 제시하여 나선형 교육이 이루어질 수 있도록 구성되어 있다.

소프트웨어교육 운영지침에서 제시된 학교급별 교육목표, 내용 요소 및 성취기준을 아래의 <표 II-7~10>에 나타내었다.

〈표 II-7〉 소프트웨어교육 학교급별 교육 목표

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 실생활의 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 기반으로 다양한 분야와 융합하여 문제를 해결할 수 있다.

〈표 II -8〉 소프트웨어교육 초등학교 내용 요소와 성취기준

영역	증영역	내용 요소	성취기준
생활과 소프트웨어	나와 소프트웨어	소프트웨어와 생활 변화	소프트웨어가 가져온 생활 모습의 변화를 설명 할 수 있다.
		사이버 공간에서의 예절	사이버 공간에서 지켜야 하는 예절을 알고 실 천할 수 있다.
	정보윤리	인터넷 중독과 예방	게임과 인터넷 중독의 문제점을 알고 예방 방법을 설명할 수 있다.
		개인 정보보호	개인 정보가 중요한 이유와 지킬 수 있는 방법을 찾아 발표할 수 있다.
		저작권 보호	생활 속에서 저작권의 보호를 받는 것은 어떤 것인지 찾아보고, 보호하기 위한 방법을 설명할 수 있다.
문제해결 과정의 체험	문제의 이해와 구조화	제시된 문제를 이해할 수 있다.	
		제시된 문제를 단순화할 수 있다.	
	문제 해결 방법 탐색	문제를 해결하기 위한 방법을 순서에 따라 설명할 수 있다.	
		제안한 문제 해결 방법의 문제점과 개선 방 법에 대해 설명할 수 있다.	
알고리즘과 프로그래밍	알고리즘의 체험	알고리즘의 개념	알고리즘의 개념을 이해할 수 있다.
		알고리즘의 체험	순차, 선택, 반복구조를 이용하여 문제해결 절차를 그림이나 기호를 이용하여 표현할 수 있다. 간단한 알고리즘(정렬, 탐색)을 체험활동을 통하여 이해할 수 있다.
프로그래밍 의 체험	프로그래밍의 이해 프로그래밍의 체험	프로그래밍 언어의 기본요소를 알 수 있다.	
		주어진 프로그램을 동일하게 만들 수 있다. 주어진 프로그램을 수정하여 자신만의 프로 그램을 만들 수 있다. 내가 생각한 간단한 프로그램을 만들 수 있다.	

<표 II -9> 소프트웨어교육 중학교 내용 요소와 성취기준

영역	중영역	내용 요소	성취기준	과정
				기본 심화
	소프트웨어의 활용과 중요성	소프트웨어의 종류와 특징	다양한 분야에서 활용되는 소프트웨어의 종류와 특징을 설명할 수 있다.	
		소프트웨어의 활용과 중요성	소프트웨어의 사용이 실생활을 어떻게 변화시켰는지 이해하고 소프트웨어의 중요성을 설명할 수 있다. 소프트웨어와 관련된 다양한 진로와 직업을 찾아보고 발표할 수 있다.	○
생활과 소프트웨어	정보윤리	개인정보 보호와 정보 보안	개인정보의 개념을 이해하고 유형을 설명할 수 있다. 개인정보 침해에 대한 피해 사례를 조사하여 예방 방법을 사례별로 설명할 수 있다.	○
		지적재산의 보호와 정보 공유	악성 프로그램의 의미를 알고 보안 소프트웨어를 설치하여 컴퓨터를 보호할 수 있다. 지적 재산의 개념을 이해하고 보호하는 방법을 설명할 수 있다.	○
		지적 재산의 보호와 정보 공유	지적 재산권을 침해하지 않고 올바른 방법으로 정보를 공유할 수 있다.	○
		컴퓨터의 구성과 정보 교류	컴퓨터의 구성 요소와 기능을 이해하고 컴퓨터가 자료를 처리하는 과정을 설명할 수 있다. 다양한 운영체제의 종류와 특징을 알 수 있다.	○
알고리즘과 프로그래밍	정보의 유형과 구조화	정보의 유형	자료와 정보의 개념을 이해하고 정보의 디지털 표현 방법을 설명할 수 있다. 정보의 유형을 구분하고 활용할 수 있다.	○
		정보의 구조화	선형 구조의 개념을 이해하고 설명할 수 있다. 비선형 구조의 개념을 이해하고 설명할 수 있다.	○
		정보의 구조화	효율적으로 정보를 제공하고 처리하기 위해서 리스트, 계층, 테이블, 다이어그램 등과 같은 다양한 형식으로 정보를 구조화할 수 있다.	○
		정보의 구조화	효율적으로 정보를 제공하고 처리하기 위해서 리스트, 계층, 테이블, 다이어그램 등과 같은 다양한 형식으로 정보를 구조화할 수 있다.	○

〈표 II -9〉 소프트웨어교육 중학교 내용 요소와 성취기준(계속)

영역	증영역	내용 요소	성취기준	과정	
				기본	심화
알고리즘과 프로그래밍	컴퓨팅 사고의 이해	문제 해결 절차의 이해	문제를 해결하는 절차를 설명할 수 있다.	○	
		문제 분석과 구조화	주어진 문제를 이해하고 분석할 수 있다.	○	
			실생활의 문제를 구조화하여 표현할 수 있다.	○	
	실생활의 문제를 추상화하여 표현할 수 있다.		○		
	알고리즘의 이해	문제 해결 전략의 탐색	주어진 문제를 해결하기 위한 다양한 해결 방법을 찾을 수 있다.	○	
		알고리즘의 이해	실생활의 사례와 연계하여 알고리즘이 무엇인지 그 의미와 중요성을 알 수 있다.	○	
			알고리즘이 갖추어야 할 조건을 이해하고 다양한 알고리즘을 비교할 수 있다.		○
		알고리즘의 설계	여러 가지의 알고리즘 표현 방법을 이해하고 설명할 수 있다.	○	
			문제 해결 절차를 여러 가지 알고리즘 표현법(자연어, 의사 코드, 순서도 등)으로 나타낼 수 있다.	○	
			실생활에서 발생하는 여러 문제들을 순차, 선택, 반복 구조를 활용하여 알고리즘을 설계할 수 있다.	○	
정렬과 탐색 등 다양한 알고리즘을 바탕으로 실생활의 여러 가지 문제를 해결할 수 있다.				○	
프로그래밍 의 이해	프로그래밍 언어의 이해	프로그래밍 언어의 개념과 종류를 설명할 수 있다	○		
	프로그래밍의 기초	프로그래밍 언어의 개발 환경을 이해할 수 있다.	○		
		자료의 입출력문을 작성할 수 있다.	○		
		문제 해결에 필요한 순차, 선택, 반복 구조를 프로그래밍할 수 있다.	○		
		여러 가지 구조적 절차를 복합적으로 구현하여 문제 해결에 활용할 수 있다.	○		
		비교-논리 연산자와 선택, 반복 구조를 활용하여 프로그래밍할 수 있다.		○	
프로그램에서 오류를 확인하여 수정할 수 있다.	○	○			
컴퓨팅과 문제해결	컴퓨팅 사고 기반의 문제 해결	실생활 문제 해결	실생활 문제를 알고리즘 설계와 프로그래밍으로 해결할 수 있다.	○	○
	다양한 영역의 문제 해결	다양한 영역의 문제 해결	다양한 영역의 문제를 알고리즘 설계와 프로그래밍으로 해결할 수 있다.	○	○

* 기본 과정 : 소프트웨어의 기본적인 역량을 학습할 수 있는 학습 요소로 구성된 운영 과정
 심화 과정 : 기본 과정을 바탕으로 컴퓨팅 사고력 기반의 문제해결 역량을 확장하기 위한 학습 요소로 구성된 운영 과정

〈표 II-10〉 소프트웨어교육 고등학교 내용 요소와 성취기준

영역	증영역	내용 요소	성취기준	
생활과 소프트웨어	컴퓨팅과 정보 생활	컴퓨팅 기술과 융합	컴퓨터 기술을 이해하고 다양한 컴퓨터 기술 분야를 찾을 수 있다. 컴퓨터 기술을 다른 학문과 융합한 예를 설명할 수 있다.	
		소프트웨어의 미래	과거와 현재의 여러 분야에서 소프트웨어가 사용됨으로써 변화된 생활의 모습을 설명할 수 있다. 다양한 분야에서 소프트웨어가 사용되는 미래 발전 방향을 예측할 수 있다.	
	정보 윤리	정보윤리와 지적 재산 보호	정보 윤리와 관련된 법과 제도를 살펴보고 정보 윤리 의식을 가질 수 있다. 지적 재산을 침해하지 않고 저작물을 올바르게 이용할 수 있다.	
		정보보안과 대응 기술	정보보안에 대한 개념을 이해하고 대응 방법을 설명할 수 있다. 정보보안 기술을 활용하여 정보를 보호할 수 있다.	
	정보기기의 동작과 정보처리	정보기기의 동작 원리	컴퓨터의 구성 요소와 동작 원리를 설명할 수 있다. 운영체제의 관리 기법을 이해하고 설명할 수 있다. 네트워크의 전송원리를 이해하고 상황에 맞게 활용할 수 있다.	
			정보처리의 과정	정보의 다양한 처리 과정을 설명할 수 있다.
	알고리즘과 프로그래밍	정보의 표현과 관리	정보의 표현	정보의 디지털 표현 원리를 설명할 수 있다. 다양한 유형의 정보를 디지털로 표현할 수 있다.
			정보의 관리	정보관리의 개념을 알고 설명할 수 있다. 정보를 효율적으로 관리할 수 있다.
		컴퓨팅 사고의 실제	문제의 구조화	문제 구조화의 개념과 방법을 설명할 수 있다. 주어진 문제를 해결하기 위해 구조화할 수 있다.
			문제의 추상화	문제 추상화의 개념과 방법을 설명할 수 있다. 주어진 문제를 추상화하여 표현할 수 있다.
모델링과 시뮬레이션			모델링과 시뮬레이션의 개념을 설명할 수 있다. 실생활의 문제를 모델링할 수 있다. 실생활의 문제를 시뮬레이션할 수 있다.	

〈표 II -10〉 소프트웨어교육 고등학교 내용 요소와 성취기준(계속)

영역	증영역	내용 요소	성취기준
알고리즘과 프로그래밍	알고리즘의 실제	복잡한 구조의 알고리즘 설계	선형구조, 비선형구조의 개념을 이해하고 알고리즘을 설계할 수 있다.
			자료를 정렬하는 다양한 방법들을 알고 알고리즘을 설계할 수 있다.
			자료를 탐색하는 다양한 방법들을 알고 알고리즘을 설계할 수 있다.
			자료를 압축하는 다양한 방법들을 알고 알고리즘을 설계할 수 있다.
		알고리즘의 분석과 평가	같은 문제를 해결하는 데 사용되는 여러 알고리즘을 비교·평가할 수 있다.
	프로그래밍 의 이해	프로그래밍 언어의 분류	절차적 언어와 객체 지향적 언어의 특징을 설명할 수 있다.
			프로그래밍 언어의 실행 절차를 설명할 수 있다.
	문제해결과 프로그래밍	프로그래밍의 실제	변수와 상수를 이해하고, 이를 이용하여 자료의 입출력 프로그램을 작성할 수 있다.
			조건문과 반복문을 이용하여 프로그램을 작성할 수 있다.
			다중 선택 구조를 이해하고 프로그램을 작성할 수 있다.
다중 반복 구조를 이해하고 프로그램을 작성할 수 있다.			
복잡한 제어 구조를 이용하여 프로그램을 작성할 수 있다.			
배열의 개념과 필요성을 설명할 수 있다 배열을 이용하여 프로그램을 작성할 수 있다. 함수의 개념과 필요성을 설명할 수 있다. 함수와 변수의 참조 범위에 대해 설명할 수 있다. 함수를 활용하여 프로그램을 작성할 수 있다.			
컴퓨팅과 문제해결	컴퓨팅 사고 기반의 융합 활동	프로그래밍과 융합	인문, 사회, 과학 분야에서 프로그래밍이 사용되는 사례를 찾아볼 수 있다.
			다양한 분야에서 문제해결과정을 찾아 이를 논리적으로 설계하고, 프로그램으로 작성할 수 있다.
	융합 활동	팀프로젝트 제작과 평가	팀을 이루어 하나의 프로젝트를 설계하고 제작할 수 있다.
			팀에서 작성한 프로그램을 발표하고 평가할 수 있다.

2) 2015 개정 실과/정보과 교육과정

2015 개정 교육과정에서의 SW교육은 운영 편제상 초등학교 실과에서 17시간, 중학교에서는 정보 과목이 필수 교과로 개편되어 34시간 교육이 이루어질 예정이다.

2015 개정 교육과정에 따르면 초등학교에서는 기존 ICT 활용 중심의 정보 단원을 SW 기초 소양 중심의 대단원으로 구성하여, 문제해결 과정과 알고리즘 및 프로그래밍 체험을 중심으로 저작권 보호 등 정보윤리 의식 함양을 포함하는 내용으로 교육을 실시하기로 하였다. 그리고 중학교에서는 전문적인 SW교육을 위해 ‘정보’ 과목을 의무화하여 컴퓨팅 사고 기반 문제해결 교육과 알고리즘 및 프로그래밍 개발을, 고등학교에서는 심화선택이던 ‘정보’ 과목을 일반선택(기술·가정 및 정보 중 한 가지 과목을 선택)으로 전환하여 다양한 분야와 융합한 알고리즘의 설계 및 프로그램 개발 중심의 교육을 실시하기로 하였다(교육부, 2015). 2015 개정 교육과정에서 제시한 학교급별 소프트웨어교육의 목표와 내용은 <표 II-11>과 같다.

<표 II-11> 2015 개정 교육과정에서의 소프트웨어교육

구분	초등학교	중학교	고등학교
교육목표	· 실과(5, 6학년) · SW 기초 소양교육	· 정보 교과 · SW 제작 원리 이해	· 정보, 정보과학 과목 · 진로와 연계, 심화 학습
교과내용	· 놀이 중심 활동 · 프로그래밍 체험	· 추상화, 알고리즘 · 문제해결과 프로그래밍	· 알고리즘과 프로그래밍 · 텍스트 기반 언어
창의적 체험 활동	· 논리적 사고 체험	· 컴퓨터 프로그램 제작	· 컴퓨팅 시스템 융합 활동

초등학교 실과 SW교육 관련 내용 요소는 기술시스템 영역에 소프트웨어 이해, 절차적 문제해결, 프로그래밍 요소와 구조로 제시되어 있으며, 성취기준 및 교수학습 방법은 <표 II-12>에 제시된 바와 같다. 한편 평가 방법 및 유의

사항으로는 ‘다양한 평가 도구를 활용하여 소프트웨어교육을 통한 컴퓨팅 사고력의 향상 정도를 측정할 수 있다’고 간략하게 언급되어 있다.

〈표 II-12〉 초등학교 실과 기술시스템 영역 내용 요소와 성취기준

영역	핵심 개념	내용 요소	성취기준
기술 시스템	소통	소프트웨어의 이해	소프트웨어가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.
		절차적 문제해결	절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.
		프로그래밍 요소와 구조	자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.
			문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.

2015 개정 교육과정 중학교 소프트웨어교육에서는 정보 문화 소양, 컴퓨팅 사고력, 협력적 문제 해결력 세 가지를 핵심 역량을 제시하고 있다. 정보 문화 소양이란 정보사회의 가치 이해, 정보사회 구성원으로서 윤리의식과 시민 의식을 갖추고 정보기술을 활용하여 문제를 해결할 수 있는 능력을, 컴퓨팅 사고력은 컴퓨터 과학의 기본 개념과 원리 및 컴퓨팅 시스템을 활용하여 실생활과 다양한 학문 분야의 문제를 이해하고 창의적으로 해법을 구현하여 적용할 수 있는 능력을, 협력적 문제 해결력이란 네트워크 컴퓨팅 환경에 기반을 둔 다양한 지식·학습 공동체에서 공유와 효율적인 의사소통, 협업을 통해 문제를 창의적으로 해결할 수 있는 능력을 의미한다. 중학교와 고등학교 정보과에 제시된 내용체계는 〈표 II-13〉과 같다.

<표 II -13> 중·고등학교 정보과 영역별 내용 요소

영역	핵심개념	내용 요소		기능
		중학교	고등학교	
정보 문화	정보사회	· 정보사회의 특성과 진로	· 정보과학과 진로	· 탐색 · 분석* · 평가**
	정보윤리	· 개인정보와 저작권 보호 · 사이버 윤리	· 정보보호와 보안 · 저작권 활용 · 사이버 윤리	
자료와 정보	자료와 정보의 표현	· 자료의 유형과 디지털 표현	· 효율적인 디지털 표현	· 분석 · 표현* · 선택**
	자료와 정보의 분석	· 자료의 수집 · 정보의 구조화	· 자료의 분석 · 정보의 관리	
문제 해결과 프로그래밍	추상화	· 문제 이해 · 핵심요소 추출	· 문제 분석 · 문제 분해와 모델링	· 비교 · 분석 · 핵심요소 추출 · 분해** · 협력
	알고리즘	· 알고리즘 이해 / 표현	· 알고리즘 설계 / 분석	
프로그래밍	프로그래밍	· 입력과 출력 · 변수와 연산 · 제어 구조 · 프로그래밍 응용	· 프로그램 개발 환경 · 변수와 자료형 · 연산자 · 표준입출력과 파일입출력 · 중첩 제어 구조 · 배열 · 함수 · 프로그래밍 응용	· 구현 · 협력
		컴퓨팅 시스템의 동작 원리	· 컴퓨팅 기기의 구성과 동작 · 운영체제 역할 · 네트워크 환경 설정	· 분석* · 활용** · 관리**
컴퓨팅 시스템	피지컬 컴퓨팅	· 센서 기반 프로그램 구현	· 피지컬 컴퓨팅 구현	· 설계

* 중학교에만 해당하는 기능

** 고등학교에만 해당하는 기능

기존의 소프트웨어교육 운영지침은 주로 교과 활동 외의 창의적 체험 활동에서 SW교육 강화를 목적으로 마련되었으나(정영식 외, 2015), 2015년에 발표된 소프트웨어교육 운영지침은 SW교육 연구학교와 선도학교에서의 수업 운영을 위한 ‘수시 개정’의 성격을 띠고 있다. 실과/정보과 2015 개정 교육과정이 고시된 이후 둘 간의 약간의 차이가 있다. 향후 교육과정 개정에 따른 변화를 예상해보기 위해 SW교육 운영지침과 2015 개정 교육과정 각론의 내용 요소와 성취기준을 중심으로 비교하여 이를 부록에 제시하였다.

다. SW교육 교재 개발 방향⁴⁾

교육부는 2015 개정 교육과정 적용 이전에 소프트웨어교육을 조기에 실시하기 위해 소프트웨어교육 운영지침을 발표하고 2015년부터 연구학교를 선정하여 적용함으로써 소프트웨어교육에 대한 인식을 확산하고 개정 교육과정 적용시 안착할 수 있도록 준비하고 있으며, 미래창조과학부는 2014년부터 소프트웨어교육 선도학교를 운영하고 있다. 교육부와 미래창조과학부는 2016년부터 선도학교 900개교를 선정하고 점차 확대함으로써 소프트웨어교육의 저변을 확대할 계획을 발표하였다(교육부·미래부, 2015).

이러한 노력에도 불구하고 초·중학교에서 소프트웨어교육 운영에 필요한 적합한 교재가 부족한 것이 사실이며, 이를 해결하기 위해 SW교육 교재 개발이 추진되고 있다. 해당 교재는 소프트웨어교육 운영지침을 초·중학교 정규 교과에서 활용하기 위해 학교급별 목표와 내용 요소, 성취기준 등을 충실히 반영하여, 2015 개정 교육과정이 적용되기 전까지 학교에서 사용될 계획이며, 학교의 여건에 맞게 SW교육이 효과적으로 운영될 수 있도록 학생용·교사용 교재를 개발하여 보급될 예정이다.

SW교육 교재는 실생활과 다른 학문 분야에서의 문제 해결에 컴퓨팅을 기반으로 문제의 이해와 분석, 자료와 정보의 수집 및 분석, 문제해결 절차의 설계 및 프로그래밍 등을 통해 컴퓨팅 사고력을 갖춘 인재를 양성할 수 있도록 개발되었다. 해당 교재는 학습 모듈을 기본 단위로 만들어져 모듈 간의 연계를 자유롭게 구성할 수 있는 방안을 모색하였으며, 전문가의 검토를 거쳐 프로토타입을 개발하고 학교 현장에 적용하였다. 또한 초등학교 실과의 정보 관련 단원이나 중학교 정보과의 내용이나 교과서와 달리 소프트웨어 중심으로 내용을 구성하고 학생들이 학습 활동을 직접 기술할 수 있는 워크북 형태로 개발하기 위해 학습 내용, 교육 환경, 교재 활용 측면에서 아래와 같은 특징을 갖도록 개발되었다.

4) 본문은 2015년 소프트웨어교육 연구학교 1차년도 성과보고회(교육부, 2015.12.03) 발표 내용을 정리한 것임.

1) 학습 내용 측면

학생의 흥미와 동기를 유발할 수 있는 의미 있는 문제 상황을 선정하여 구성하며, 학생들이 문제를 해결하는 과정 및 결과를 기록관리할 수 있는 워크북 형태로 개발하였다. 교재는 문제 해결 과정 중심으로 개발하기 위해 문제 제시, 문제의 이해와 분석, 문제 해결 방법 탐색, 알고리즘 설계, 프로그래밍, 검증 등 컴퓨팅 사고력의 구현 절차를 일련의 활동으로 수행할 수 있도록 내용을 구성하여 개발하였다.

학습자 활동 중심의 교재로 학습 보조 자료도 함께 제시하여 학습자들이 직접 활동의 결과를 기록하고 관리할 수 있는 형태로 개발한다. 수학, 과학 등의 타 교과 학습 요소를 반영한 문제 상황을 제시하여 융합 활동을 유도하고자 하였다. 학습 상황에 따라 블록형 프로그래밍 언어(스크래치, 엔트리 등)와 텍스트 기반의 언어(파이썬, C 언어 등)가 모두 가능하도록 내용을 구성하여 다양한 프로그래밍 도구의 사용을 고려하였다.

2) 교육 환경 측면

동일한 학습 요소를 대상으로 컴퓨터 사용이 가능한 환경과 컴퓨터 사용이 어려운 환경에서도 학습 활동이 가능하도록 다양한 교육환경을 고려하여 내용을 구성하였다. 학습자 수준 및 선택 시수의 다양성 고려하되, 중학교의 경우 기본심화과정을 반영한 모듈형 교재를 개발하고, 초등학교의 경우에는 확보된 시간 외에 창의적 체험활동 등을 통해 컴퓨팅 사고력을 체험해 볼 수 있는 다양한 활동 내용으로 구성하였다.

소프트웨어교육 교재의 수업 시수는 학습 수준별 교육이 가능하도록 초등학교는 17차시(기본 17차시, 심화 17차시), 중학교는 68차시(기본 34차시, 심화 34차시) 분량으로 개발되었다. 그리고 모듈별 시수는 <표 II-14>와 같다.

〈표 II-14〉 SW교육 교재의 모듈별 시수

구분	교재 시수		모듈별 시수 (모듈별 수)			
	기본	심화	지식모듈	활동모듈	기능모듈	계
초등학교	17차시	17차시	6차시 (4개)	25차시 (9개)	3차시 (3개)	34차시 (16개)
중학교	34차시	34차시	15차시 (15개)	48차시 (16개)	5차시 (5개)	68차시 (36개)

3) 교재 활용 측면

학생이 직접 워크북에 학습 활동을 기록함으로써 학습과정 완료시에는 교재가 학생의 학습결과물이 될 수 있도록 개발하였다. 교사의 역량과 학습자의 수준 등을 고려하여 교사들이 쉽게 내용을 재구성하여 활용할 수 있도록 학습 내용을 모듈 단위로 개발하였다. 교과 특성 반영하여 단순한 오프라인 교재가 아니라 온라인 플랫폼(에듀넷 등)과 연동한 형태로 개발하였다. 학습 활동 모듈은 연구학교 교사들이 지속적으로 추가 개발하여 교사들 상호간에 공유하여 활용할 수 있으며, 편집이 가능한 형태로 개발하였다.

〈표 II-15〉 초등학교 모듈 구성안

유형	내용 요소(기본 17차시)+(심화 17차시)				
	소프트웨어 활용과 중요성	정보 윤리	문제 해결 방법 찾기	알고리즘 설계	프로그래밍
지식 모듈	K-1	K-2	K-3	K-4	
기능 모듈					P
활동 모듈	주제-A		A-3	A-4	A-P
	주제-B		B-3	B-4	B-P
	주제-n		n-3	n-4	n-P

〈표 II-16〉 중학교 모듈 구성안

유형	내용 요소(기본 17차시)+(심화 17차시)						
	소프트웨어 활용과 중요성	정보 윤리	정보 유형	문제 해결 절차	문제해결 방법찾기	알고리 즘 설계	프로 그래밍
지식 모듈	K-1	K-2	K-3	K-4	K-5	K-6	
기능 모듈							P
활동 모듈	주제-A				A-5	A-6	A-P
	주제-B				B-5	B-6	B-P
	주제-n				n-5	n-6	n-P

모듈은 서로 독립적으로 학습할 수 있으며, 교사들이 확보된 시간이나 학습자의 수준에 맞추어 알맞게 재구성하여 활용할 수 있다. 단, 프로그래밍 모듈은 반드시 알고리즘 모듈과 연계하여 구성하여야 한다. 아래는 중학교 모듈 재구성 예시이다.

- 지식 모듈을 먼저 학습하고 활동 모듈을 학습하는 구성
 〈지식모듈 1-6〉 → 〈활동 주제-A 학습〉 → 〈활동 주제-B 학습〉
- 활동 모듈을 먼저 학습하면서 지식 모듈과 연계하여 학습
 〈A-5〉 → 〈K-5〉 → 〈A-6〉 → 〈K-6〉 → 〈A-P〉
- 여러 주제의 문제해결 방법찾기 모듈을 먼저 학습한 후, 특정 주제의 알고리즘 설계와 프로그래밍을 학습하는 구성
 〈A-5〉 → 〈B-5〉 → 〈C-5〉 → 〈A-6〉 → 〈A-P〉 → 〈B-6〉 …

소프트웨어교육 교재의 내용 체계는 「소프트웨어 교육 운영 지침」의 중영역과 내용 요소를 중심으로 재구성하여 학습 주제를 선정하고, 교재 내용 특성에 따라 지식·활동·기능의 세 가지 유형으로 구분하여 학습 주제를 분류하였으며, 세 유형은 지식모듈, 활동모듈, 기능모듈로 설정하였으며, 각 모듈의 의미는 아래와 같다.

- 지식모듈(Knowledge Module) : 해당 영역의 이론과 지식을 학습하는 내용
- 활동 모듈(Action Module) : 학습한 내용을 토대로 실제 문제 상황을 문제해결 방법찾기 → 알고리즘 설계 → 프로그래밍 일련의 절차로 해결하는 내용
- 기능 모듈(Programming Module) : 프로그램 개발에 필요한 프로그래밍 기능을 가르치는 내용

〈표 II -17〉 지식·활동 모듈의 구성 요소

지식 모듈 요소	활동 모듈 요소		
	활동1 문제해결 방법찾기	활동2 알고리즘 설계 해결방법 설계하기	활동3 프로그래밍
① 학습목표	① 생각 열기	① 문제해결 방법 표현하기	① 준비하기
② 학습활동	② 생각 모으기	② 알고리즘 탐색1	② 프로그래밍하기
③ 개념의 구조화	③ 생각 다듬기	③ 알고리즘 탐색2	③ 실행하기
④ 개념 설명	④ 생각 나누기	④ 알고리즘 설계	④ 더 나아가기
⑤ 예제 및 사례			

소프트웨어교육 교재의 각 모듈별 예시 화면⁵⁾을 제시하면 [그림 II -1~7]과 같다.

5) 교육부와 한국교육학술정보원에서 개발하고 있는 SW교육 학생용 교재에서 발췌. in press.

1 문제 해결 방법을 표현해요.

이 학습 내용을 공부하고 나면 알고리즘의 뜻을 설명할 수 있어요.

알고리즘 이해하기

컴퓨터로 문제를 해결하려면 어떻게 해야 할까요? 혹은 문제를 해결하는 방법을 다른 사람에게는 어떻게 전달할 수 있을까요? 문제 해결 방법이나 절차를 컴퓨터, 혹은 다른 사람들이 이해할 수 있도록 표현하는 방식에 대하여 알아봅시다.

다음 그림을 보면서 문제 해결 방법을 어떻게 표현할지 생각해 봅시다.

나 중요한 정보를 빠뜨리지 않을려면?

효율적인 해결책이 있다고 하는 방법을 기록할 수 있을까?

올라가는 순서대로 차례에 기록해 주면 되지는 않을까?

그걸 어떤 방법으로 표현할 수 있을까?

의사 순서도와 선형식 로봇 프로그램 제작 시 나가는 과정을 수행하는 모습!

여러하는 방법을 사용하는 방법! 순서대로 기록해 주어야 합니다!

알고리즘이란 무엇인가요?

알고리즘이란 문제를 해결하는 절차나 방법을 순서대로 나열한 것을 말합니다. 알고리즘을 표현하는 방법에는 여러 가지가 있지만 대표적인 방법 중의 하나가 순서도입니다.

왼쪽 그림은 순서도로 알고리즘을 나타낸 것이고, 오른쪽 그림은 왼쪽 순서도로 나타낸 알고리즘을 실행하기 전과 후의 상태를 표현한 것입니다.



왼쪽에 있는 순서도에 따라 실행했을 때 상자 A와 상자 B에는 어떤 공이 남아 있을까요?

읽을 거리

순서도에서 사용하는 기본적인 기호는 다음과 같습니다.

기호	이름	설명
○	단일	순서도의 처음과 끝을 표시합니다.
□	처리	문제 해결을 명령을 실행합니다.
◇	판단	조건에 따라 다른 명령을 실행합니다.
▭	반복	지정된 만큼 명령을 반복하여 실행합니다.
▭	입출력	입력/출력을 표시합니다.
→	흐름	명령의 흐름을 표시합니다.

[그림 II-1] 지식 모듈 예시

6
7

달려라 거북이!

나는 왜 이렇게 느리!

어느 날, 거북이가 영급영급 기어가고 있었습니다. 토끼가 경주장중 뛰어가다가 거북이를 보고 말했 습니다. "거북아, 너는 정말 걸음이 느리구나. 너는 빨리 걸음을 못하냐?" 토끼는 거북이를 놀렸습니다.

오브젝트 상태, 추가하기

토끼, 거북이, 돌판 오브젝트를 추가해 봅시다.

1 엔트리뷰트 오브젝트를 클릭하여 선택해요. '오브젝트 추가하기'를 클릭하여 '토끼-돌판', '돌판-토끼, 거북이' 오브젝트를 추가하고 '제출하기'를 선택해요.

2 돌판 4, 토끼, 거북이를 위쪽에서 같이 배치해요.

오브젝트 움직이기

토끼 오브젝트를 움직여 봅시다.

1 '시력' - 시력 버튼을 클릭했을 때 블록을 끌어서 블록 조합소에 가지고 와요.

2 '조건' - 1회만 반복하기 블록을 가져와서 '시력 버튼을 클릭했을 때' 블록 아래에 붙여요.

3 '움직이기' - 이동 방향으로 (10만큼 움직이기) 블록을 가져와 100번 반복하기 블록의 안에 붙여요.

4 '영양' 블록

영양이 블록은 끌어서 조합해요. 영양이 블록을 끌어와서 위에서 끌어와서 블록 조합소에 가지고 와요. 다른 블록을 사용해 때, 기존 영양이 블록이 필요하기만 블록이 사용으로 붙여주세요. 때이고 싶은 블록도 클리킹 상태로 끌어와서 배치 하면 돼요.

블록은 여러 번 사용할 수 있어요. 블록 꾸러미에 있는 블록은 조합으로 가지고 와서 여러 번 사용할 수 있어요.

[그림 II-2] 기능 모듈 예시

로봇으로 교실 청소하자!

1

활동 목표
청소 로봇 프로그램을 만들 수 있다.

활동 내용

- 청소 로봇이 움직이는 과정을 역할 놀이로 이해하고, 청소 로봇이 이용하는 방법 표현하기
- 순차 반복에 따라 실행하는 알고리즘을 설계하고 표현하기
- 교실을 청소하는 로봇 프로그램 만들기

※ 활동 학습을 시작하기 전, 알고 있는 내용을 체크해 봅시다.

· 청소 로봇이 어떻게 움직이는지 알고 있다.	<input type="checkbox"/>
· 순서대로 실행하는 프로그램을 이해하고 있다.	<input type="checkbox"/>
· 반복에 따라 실행하는 프로그램을 이해하고 있다.	<input type="checkbox"/>

2

활동 주제 | **로봇으로 교실 청소하자!** 문제 해결 방법 찾기

문제 해결 방법 찾기

이 이야기에서 해결해야 할 문제가 무엇일까요?

1 청소하기 너무 힘들어.....

2 청소할 때가 대신 해 주면 좋을까?

3 로봇이 청소 해 주면 어떤가?

4 간단한 청소 로봇 프로그램을 한 번 만들어 보는 건 어때?

5 "아! 먼저 청소 로봇이 어떻게 움직이는지 알아보자."

6 "순차 반복된 모양의 바닥에서 한 칸씩 움직이면서 청소를 하는 로봇이 있다고 생각하면 좋을 거 같아."

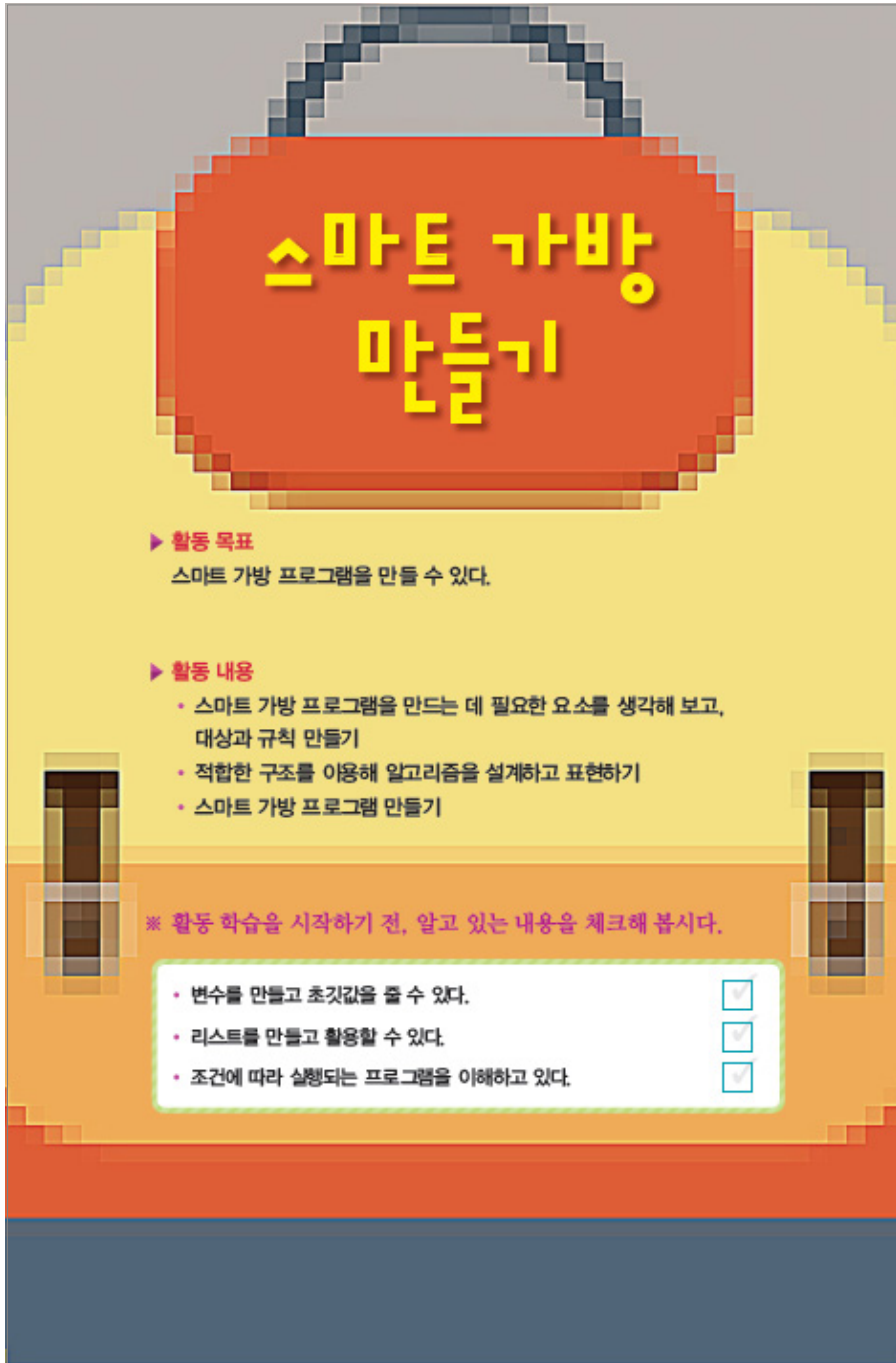
7 "좋은 생각이네. 우리 교실은 직사각형 모양이니까 로봇이 어떻게 움직이면서 청소를 할 수 있을지 생각해 볼까?"

8 "그 다음엔 교실 바닥을 청소하는 로봇 프로그램을 어떻게 만들지 생각해 보지요~"

위 대화를 읽고 해결해야 할 문제를 정리하여 아래 빈칸에 채워 봅시다.

누가 청소하는 학생	언제 정과 후	어디에서 교실에서	왜 청소하기 힘들기 때문
어떻게	무엇을		
.....		
.....		

[그림 II-3] 활동 모듈 예시



[그림 II-4] 활동 모듈 개요 페이지 예시

2 문제 해결 방법 찾기 문제 해결 방법 찾기

문제 해결 방법 찾기

이 이야기에서 해결해야 할 문제가 무엇일까요?

❑ 위 대화를 읽고 해결해야 할 문제를 정리하여 아래 빈칸에 채워 봅시다.

누가	언제	어디에서	왜
청소하는 학생	방과 후	교실에서	청소하기 힘들기 때문에

어떻게	방법을
알아보고	교실 바닥을 청소할 수 있는 방법을 찾아보고, 문제를 해결한다.

3 문제 해결 방법을 생각해 보아요. 문제 해결 방법 찾기

문제 해결 방법을 생각해 보아요.

❑ 실편 앞의 출발점까지 도착점까지 이동하는 로봇 역할 놀이를 해 봅시다.

역할 놀이 규칙

- 1 두 사람이 책이 되어 로봇과 조종사 역할을 맡는다.
- 2 로봇이 앞에, 조종사가 뒤에 선다.
- 3 로봇은 조종사의 명령에 따라 움직인다.
- 4 조종사가 내릴 수 있는 명령은 "앞으로 한 칸을 가기", "오른쪽으로 90도만큼 돌기", "왼쪽으로 90도만큼 돌기" 세 가지이다.
- 5 조종사는 한 번에 한 가지 명령만 내릴 수 있다.

❑ 로봇 역할 놀이를 해 보고, 청소 로봇 프로그램을 어떻게 만들어야 할지 예시 찾아 두 그림이 안에 표시해 봅시다.

움직이는 방법

앞으로 (한 칸 가기) 오른쪽으로 (90도 돌기) 왼쪽으로 (90도 돌기)

TIP 한 번에 이동할 수 있는 거리는 한 칸입니다. 청소 로봇은 한 번에 한 가지 명령만 실행합니다.

❑ 청소 로봇(●)이 있거나(?)이 있는 곳까지 청소하는 경우

❑ 청소 로봇이 가로, 세로 2칸인 교실 바닥을 모두 지나며 청소하는 경우

❗ 중요한 내용을 다시 한번 살펴봐요!

[그림 II-5] 문제해결 방법찾기 예시

5 알고리즘 만들기 알고리즘 만들기

알고리즘 만들기

알고리즘을 살펴보아요.

교실 바닥을 청소하는 청소 로봇 알고리즘을 생각해 봅시다.

❑ 청소 로봇이 움직이는 방법에 대해 생각해 보고, 일련성에 들어갈 말을 **모기**에서 찾아 써 봅시다.

모기

- 벽
- 앞으로 한 칸 가기
- 오른쪽으로 90도만큼 돌기

어떻게 하자? 이거 있었어? 청소 로봇이 더 이상 앞으로 갈 수 없을 때는? 청소 로봇이 다른 방향으로 가야 합니다.

다 이상 앞으로 갈 수 없으니 돌(을) 해야겠다.

❑ 청소 로봇이 움직이는 방법에 따라 청소를 하려고 합니다. 움직이는 방법을 참고하여 청소 로봇을 움직여 봅시다.

움직이는 방법

앞으로 (한 칸 가기) 오른쪽으로 (90도 돌기) 왼쪽으로 (90도 돌기)

TIP 한 번에 이동할 수 있는 거리는 한 칸입니다. 청소 로봇은 한 번에 한 가지 명령만 실행합니다.

6 알고리즘 만들기 알고리즘 만들기

❑ 위 활동을 참고하여, 순서대로 실행하는 알고리즘과, 반복에 따라 실행하는 알고리즘을 이용해 봅시다.

❑ 순서대로 실행하는 알고리즘 이해하기

모기와 같이 청소 로봇(●)이 움직이는 방법은 방법에 따라 교실 바닥을 모두 청소하고 제자리로 돌아오도록 합니다. 청소 로봇 프로그램을 어떻게 만들어야 할지 **모기**를 보고 순서대로 아래의 빈칸에 써 봅시다.

그림 순서대로 움직여요.

모기

- 앞으로 한 칸 가기
- 오른쪽으로 90도만큼 돌기
- 오른쪽으로 90도만큼 돌기

오른쪽으로 90도만큼 돌기

앞으로 한 칸 가기

오른쪽으로 90도만큼 돌기

앞으로 한 칸 가기

오른쪽으로 90도만큼 돌기

앞으로 한 칸 가기

오른쪽으로 90도만큼 돌기

앞으로 한 칸 가기

오른쪽으로 90도만큼 돌기

앞으로 한 칸 가기

오른쪽으로 90도만큼 돌기

[그림 II-6] 알고리즘 만들기 예시



[그림 II-7] 프로그래밍하기 예시(스크래치/엔트리를 같이 제공)

소프트웨어교육 초등학교 교재에는 스크래치와 엔트리를 이용한 프로그래밍 과정이 같이 제공되어 선택의 폭을 넓히고자 하였으며, 중학교 교재에는 이외에도 파이썬 프로그래밍 과정도 포함되어 있다.

3. 교수학습 모델 선행연구 분석

프로그래밍 모델, 응용 SW 모델, 멀티미디어 모델, 문제중심모델 등 지금까지 정보 교과에 제안된 교수학습 모델은 무수히 많다. ICT 활용이 강조된 2000년대 이후에는 도구적 활용 방안, PBL과 같은 기존 교수-학습 모델에 ICT를 접목하는 방안, 새로운 교수-학습 모델 제안 등이 이어졌다. 이는 정보 교과 교수학습 모델 또는 방법이 명확하게 정해져 있다기보다 상황에 맞춰 달라질 수 있음을 시사한다(이태욱, 유인환, 이철현, 2001; 이태욱, 최현중, 2015).

다시 말해서 교수학습 방법은 어떤 특정한 교수학습 모델을 특정하기보다, 교수 목적, 학습 내용의 특성, 학생의 특성, 학습환경 등을 고려하여 적절하게 선택할 필요가 있다는 의미이다. 컴퓨팅 사고력이 일종의 문제해결 능력으로 강조되고 있다는 점에 중점을 두고 문제중심학습과 문제해결모델을, 최근 컴퓨팅 사고력과 관련된 연구가 많이 보고된 디자인기반학습 모델을, 또한 학습자 중심의 교수학습방법으로 플립러닝을 기반으로 한 S/W 교육방법에 대하여 개괄적으로 살펴보면 다음과 같다.

가. 문제 중심 학습(Problem Based Learning, PBL)

PBL은 실제로 발생하는 문제와 상황을 중심으로 교수-학습을 구조화한 교육적 접근으로서, 학습자들이 문제를 협력적이고 자기주도적으로 해결해 가는 과정을 통해서 내용에 대한 학습, 비판적 사고력과 협력기능을 기르도록 하는 교수학습 형태이다(김신자 외, 2000). Barrows와 Tamblyn(1980)은 문제중심 학습을 ‘문제에 대한 이해 혹은 문제해결을 위해 이루어지는 활동과정에서 산출되는 학습이다’라고 정의한 바 있다.

문제중심학습에서 교수자는 학습자들을 가르치는 것이 아닌 학습자가 비판적으로 사고하고, 자기주도적으로 학습하는 과정을 통해 스스로 내용지식을 생성할 수 있도록 지원하는 입장에 있다. 이러한 PBL은 의과대학 교수로 있던 Barrows와 Tamblyn(1980)에 의해 구체화되어 경영학, 교육학, 사회학 그리고 건축학과 공학 등 대학교육 뿐만 아니라 고등학교에도 폭넓게 적용되어 온 학습법이다(Savery & Duffy, 1996).

PBL은 교수학습 결과 창의적 사고력, 비판적 사고력, 문제해결력, 협동 능력 등을 목표로 하고 있다(조연순, 2006). 이러한 목표는 소프트웨어교육에서 강조하고 있는 컴퓨팅 사고력을 바탕으로 실생활의 문제를 해결하는 과정의 경험과 창의적인 문제해결(교육부, 2015)와도 상당 부분 중첩되어 있어 SW교육의 여러 수업 활동에 PBL을 적용할 수 있을 것이라 예상된다. 예를 들어 교육용

프로그래밍언어(Education Programming Language, EPL) 수업에서 문법 및 개념을 직접 교수하는 방법도 있지만, 간단한 문제나 예시를 함께 만들어 보며 지식과 기능을 자연스럽게 익히도록 할 수도 있을 것이다. 일반적인 PBL의 수행 과정을 문제 확인, 문제 해결 계획 수립, 탐색 및 재탐색, 해결책 고안, 발표 및 평가의 다섯 단계로 구분하고, 이 따른 교수학습 과정 및 목표로 하는 바를 [그림 II-8]에 정리하였다.

단계	문제 확인	문제 해결 계획 수립	탐색 및 재탐색	해결책 고안	발표 및 평가
교수-학습 과정	<ul style="list-style-type: none"> • 동기 유발 • 문제 제시 • 문제 파악 	<ul style="list-style-type: none"> • 계획 수립 - 알고 있는 것 - 알아야 할 것 - 알아내는 방법 	<ul style="list-style-type: none"> • 지식·정보의 수집 • 탐색과정 반복 	<ul style="list-style-type: none"> • 해결책 고안 및 도출 	<ul style="list-style-type: none"> • 해결책 공유 및 평가 • 반성 및 성찰
기르고자 하는 능력	문제발견능력	창의적 사고력 비판적 사고력	전문적 지식 비판적 사고력	창의적문제해결력 의사소통 및 협동 능력	의사결정능력

[그림 II-8] 문제중심학습의 일반적 절차(조연순, 2006을 재구성)

나. 문제 해결 모델

문제 해결 모델은 학습자 주도의 문제 해결 과정을 강조하는 학습자 중심의 교수학습모델이다. 이 모델은 최대한 학습자 스스로 문제 해결 방법을 찾아 문제를 해결하도록 유도함으로써 자발적인 학습 참여를 유도하고 학습자의 탐구력을 신장시키는 데 유리하다. 학습자는 문제 해결 과정에서 지식이나 개념을 단순 수용하는 것이 아니라 나름대로 지식이나 개념을 탐구하고 재구성할 수 있는 기회를 가짐으로써 학습에 대한 주도성을 가지게 된다(Sternberg, 1996; 강숙희, 1998에서 재인용).

ICT 활용에 중점을 두고 정보를 다루는 기술(skills)을 일반적으로 정보기법이라 하며, Big 6 기술이 정보기법을 기반으로 하는 가장 일반적인 모델이다.

이 모델은 대부분의 문제 해결 모델에서 제시되는 기본적이고도 중요한 단계를 포함하고 있으며, 정보와 관련된 문제를 해결하기 위한 모델이라는 설계상의 특징을 지닌다(이태욱 외, 2001). 문제 해결 모델의 6가지 단계별 세부 요소를 기술하면 <표 II-18>과 같다.

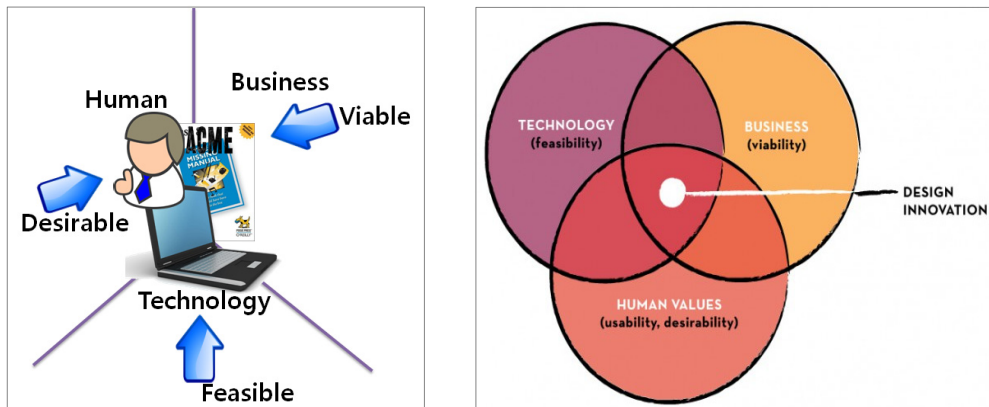
<표 II-18> The Big 6 Skills

단계	요소
과제 확인	1.1 문제 정의 1.2 과제를 해결하는 데 필요한 정보를 규명(identify)
정보 탐색 전략	2.1 가능한 자원(sources/brainstorm)의 범위 결정 2.2 서로 다른 가능한 자원을 평가하고 최선의 것을 선택
검색	3.1 정보 자원의 위치 알아내기(intellectually and physically) 3.2 자원 안에서 정보 찾기
정보 사용	4.1 자원에서 정보를 읽고, 보고, 듣고, 느끼기 4.2 자원으로부터 관련된 정보 추출
종합	5.1 다양한 자원으로부터 정보 조직 5.2 정보 제시
평가	6.1 결과 평가(효과성) 6.2 정보 문제 해결 과정 평가(효율성)

이러한 문제해결모델은 학습자들에게 실생활에서 일어날 수 있는 실제적 문제를 제시함으로써 그 문제를 해결하는 과정을 강조하고 있다. SW교육이 프로그래밍 기술의 습득보다 컴퓨팅 사고력의 증진에 주안점을 두고 있다는 점에서, 실생활의 다양한 문제를 컴퓨팅으로 해결하는 경험을 제공하는 과정이라는 측면에서 활용 가능한 모델이라 할 수 있다.

다. 디자인 기반 학습(Design Based Learning, DBL)

디자인 기반 학습을 이해하기 위해서는 디자인 사고의 개념을 살펴볼 필요가 있다. 디자인 사고는 지난 10여 년 간 비즈니스 분야에서 크게 유행한 문제해결방법론이자, 교육 실행 도구의 하나이다. 디자인 기업으로 유명한 IDEO에서는 디자인 사고란 고객의 가치와 시장의 기회를 기술적으로 가능한 성공적인 경영 전략으로 전환하기 위한 요구를 충족시키기 위해 디자이너의 감수성과 방식(method)을 이용하는 실행 방법이라고 묘사한 바 있다(IDEO's Blog, 2008). 또한 Martin(2009)은 기존 분석적 사고의 중심이 연역과 귀납인데 반해 디자인 사고는 귀추법(abductive reasoning)과 통합적 사고를 강조하는 사고 방법이라고 정의한 바 있다. 스탠포드 대학교에서는 디자인 사고를 교육에 적용한 D-school⁶⁾ 개념을 제안하면서, '교육자에게 디자인 사고란 교실, 학교, 지역 사회에서 의미 있는 해결책을 설계하는 것'이라고 정의하였다.



[그림 II-9] 디자인 사고의 개념⁷⁾

6) 디자인 사고 k-12 educator's toolkit(<http://www.designthinkingforeducators.com/>).

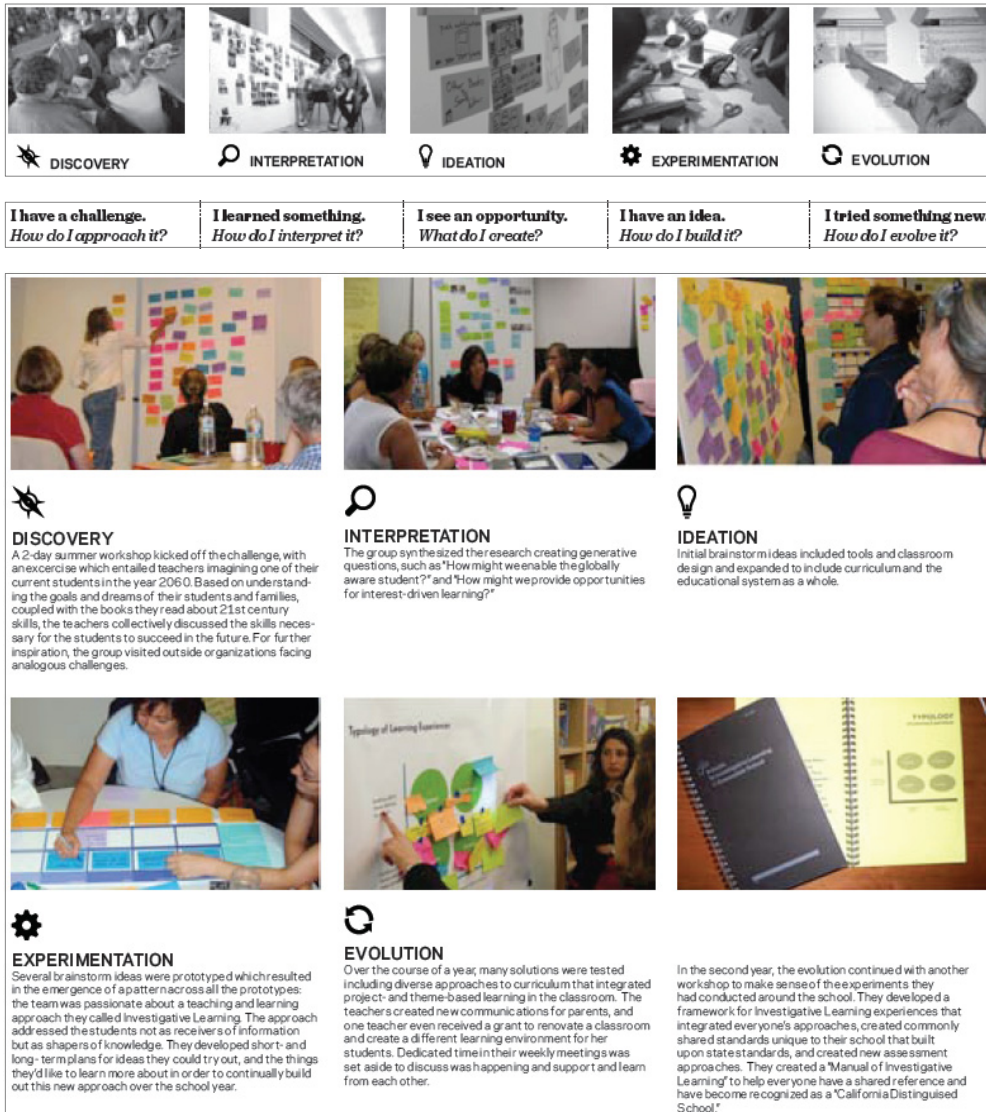
7) 출처 : IDEO's Blog(<http://designthinking.ideo.com/>),
D-school(<http://dschool.stanford.edu/>).

특히, D-school의 디자인 기반 학습은 주로 과학교육의 탐구학습과 비교되는데, 기존 탐구학습의 경우 실험을 통한 가설 검증이 이루어지는 반면에 디자인 기반 학습에서는 아이디어를 직접 제작하고 테스트 하는 활동을 중심으로 이루어진다. 이러한 D-school의 교육방법은 공감하기, 정의하기, 상상하기, 프로토타입, 테스트의 다섯 절차를 거친다. 각 단계별 세부 활동의 예시를 살펴보면, 공감하기에서는 관찰하고, 그 일에 직접 관여하고, 경험해 보는 과정으로, 실제 DBL이 표출되는 부분은 상상한 것을 프로토타입 형태로 제작하여 테스트하는 과정으로 구성되어 있다(김수환, 한선관, 2012). 기존의 과학 탐구 활동과 D-school의 단계를 비교하면 <표 II-19>와 같다.

<표 II-19> 디자인 기반 학습의 단계(김수환, 한선관, 2012)

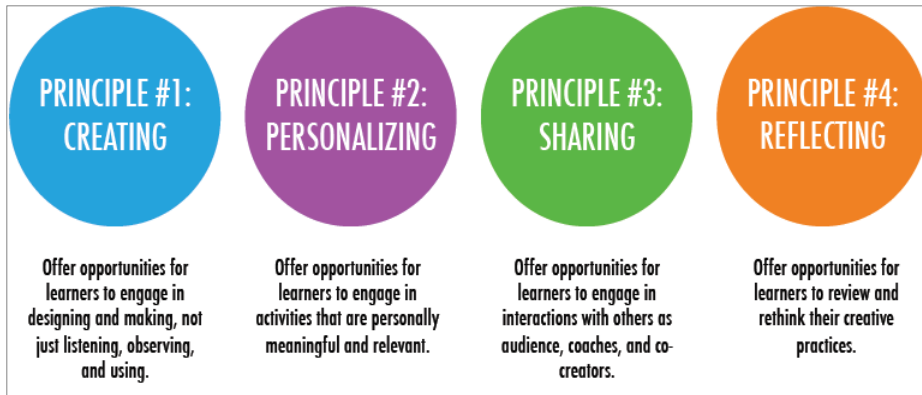
구분	과학실험 (IBL)	21st Skill (DBL)	D-school	CL교육 (DBL)	과학창의재단 (STEAM)
	질문 제기	문제 정의	공감하기 (Empathy)	계획하기	상황제시
	고민	문제 고민	정의하기 (Define)	연결하기	
	가설 설정	설계, 기획, 프로토타입 도출	상상하기 (Ideate)	탐색하기	창의적 설계
단계	실험 검증	프로토타입, 해결책 찾기	프로토타입 (Prototype)	제작하기	감성적 체험
	실험결과 분석	결과 분석	테스트 (Test)	되돌아보기	보상, 새로운 문제 도전
	결과 비교	공유, 시장에 출시			
	발전 및 새로운 질문	더 나은 프로토타입, 새로운 문제			

한편 스탠포드대학의 D-school에서는 교육 분야에 디자인 사고를 적용하기 위한 활용 도구(toolkit)을 제공하고 있다(IDEO & Riverdale, 2012). 활용 도구에서 제안하고 있는 교수학습 절차는 [그림 II-10]과 같다.



[그림 II-10] 디자인 사고 교수학습 절차

이 밖에도 MIT 미디어랩은 창의 컴퓨팅 가이드북에서 DBL 교육 프로그램을 제안하였는데, Brennen, Balch와 Chung(2014)은 창의 컴퓨팅이란 창의성, 역량의 강화, 컴퓨팅에 관련된 것으로, 문제를 해결하기 위해 디자인, 컴퓨팅, 융합의 과정을 거쳐 창의적으로 문제를 해결할 수 있는 소프트웨어를 산출하는 활동이라고 정의하였다. 창의 컴퓨팅 교육 프로그램은 컴퓨팅 사고를 바탕으로 디자인하기(무언가를 사용하는 것이 아니라 창조하고 상호작용하는 것), 개인화하기(개인적으로 의미 있고 연관된 것을 창조하는 것), 협력하기(창조물을 다른 사람과 함께 만드는 것), 되돌아보기(창조물을 되돌아보고 다시 생각해 보는 것) 네 단계로 제시되어 있다(Brennen 외, 2014; 김수환, 2011).



[그림 II-11] 창의 컴퓨팅 디자인 기반 학습 절차(Brennen 외, 2014)

한편 한선관(2015)은 창의 컴퓨팅을 학교 수업에 적용하기 위한 교수학습 모델을 초등 저학년용은 U-M-C(Use, Modify, Create), 초등 고학년용은 D-D-C(Discovery, Design, Create), 중등용은 D-C-C(Design, Computing, Convergence) 모델 3가지로 제안하였다. 각 모델별 활동을 <표 II-20~22>에 제시하였다. 또한 Brennen과 Resnick(2012)의 제안에 기반을 두어 개념 평가, 프로젝트 기반 산출물 평가, 프로젝트 포트폴리오 분석, 인터뷰, 디자인 시나리오, 지필평가, 자기평가/동료평가의 평가방법을 제시하였다.

〈표 II -20〉 UMC 모델

단계	설명	활동 예시
Use (Play)	해보기	공 튀기기 게임을 실행해 봅니다. 공이 튀기는 각도가 어떻게 작동하는지 생각해 봅니다. 공과 바의 관계를 생각해 봅니다.
Modify	수정 확장 보완	공과 바를 만들어봅니다. 벽돌을 추가합니다. 공과 바의 관계를 생각하고, 이와 같은 패턴을 공과 벽돌에 적용해 봅니다. 점수를 추가해 봅니다.
Create	배운 것을 토대로 새 산출물	공 튀기기 게임을 응용하여 다양한 게임을 만들어 봅니다. 물고기 게임, 상어 게임, 공 튀기기 업그레이드 버전 등

〈표 II -21〉 DDC 모델

단계	설명	활동 예시
Discovery	<ul style="list-style-type: none"> · 도전(챌린지) · 디버깅(주어진 문제의 문제점을 발견하고 수정하면서 기능, 원리, 개념을 익힘) · 컴퓨팅 활동 	<ul style="list-style-type: none"> - 미래 에너지 문제 스토리 - 도시에 필요한 전기량과 관련된 스토리 - 풍력발전소 시뮬레이터 만들기 목표설정 - 실제 풍력발전소의 1일 전기량 조사하기 - 도시의 전력 소비량 조사하기
Design	<ul style="list-style-type: none"> · CT 디자인 (문제분해-추상화-협력-아이디어확정) · Design Thinking (강조-정의-창의설계-프로토타입-공유평가) 	<ul style="list-style-type: none"> - 시뮬레이터 구상하기 - 시뮬레이터 프로그램 순서도 작성하기
Create	<ul style="list-style-type: none"> · 배운 것을 토대로 새로운 산출물 생산 · 컴퓨팅과 융합 	<ul style="list-style-type: none"> - 풍력발전소 시뮬레이터 제작하기(프로그래밍) - 그룹별로 협력하기 - 테스트와 디버깅하기 - 시뮬레이터 발표 및 공유하기 - 다른 그룹의 작품을 보고 비교해보기 - 자신들의 시뮬레이터를 수정하기 - 최종 작품을 공유하기

〈표 II -22〉 DCC 모델

단계	설명	활동 예시
Design	디자인	- 질병문제, DNA 구조, 질병 발견 방법 - DNA 구조 모델 맞추기
Computing	인플러그드 EPL 피지컬	- 리스트를 이용한 텍스트 매칭 프로그래밍 - DNA 구조 검출 프로그래밍
Convergence	융합	- DNA와 질병 검출 - DNA 시퀀싱 원리 - DNA 질병 검출 프로그램 설계

〈표 II -23〉 창의 컴퓨팅 평가방법

평가방법	세부 내용
개념평가	- 변수, 연산자, 반복과 같은 CS 개념과 슈도코드, 프로그래밍 개발(코딩 교육) 평가 - 디버깅 문제
프로젝트 기반 산출물 평가	- 프로젝트 기반의 산출물 평가는 장기 프로젝트에서 가능 - 학기말 과제, 팀 프로젝트, 산출물 대회 등
프로젝트 포트폴리오 분석	- Brennen의 분석방법으로 온라인에 탑재된 프로젝트 포트폴리오를 color search engine으로 분석 - 디자인과 확장된 부분에서는 아이디어 노트, 즉 디자인 저널을 통해 다면적 평가
인터뷰	- 산출 과정 인터뷰 - 창작 소감 및 태도(협력 등), 프로젝트 창작, 온라인 커뮤니티, 앞으로의 활용 → 평가 루브릭, 워크시트 개발 후 적용 - 디자인 아이디어에 대한 창의성
디자인 시나리오	- 프로젝트 명칭, 퍼포먼스 정하기 → 예제 시나리오를 만들거나 수정, 보완 - 실제 개발 없이 개발된 작품을 학생에게 보여주고 개발 시나리오와 프로그래밍 기법, 알고리즘 등에 대해 설명하는 것 등 - 의도적으로 오류가 포함된 문제를 제시하고, 그에 대한 해결(디버깅이 필요한 경우 그에 대한 문제해결방법을 설명하게 함)
지필평가	- 프로그램 문법, 정보 올림피아드 문제, 디버깅 문제 등
자기/동료평가	- 참여에 대한 체크리스트, 산출물에 대한 동료 평가표 등

한편 최형신(2014)은 교육대학 수업 및 평가 방법을 위한 연구를 통해 Brennen과 Resnick(2012)의 CT 평가 프레임워크를 바탕으로 컴퓨팅 사고력 평가 방안을 제시한 바 있다. 제안된 평가 루브릭을 아래의 표 <II-24>에 나타내었다.

<표 II-24> CT 역량 평가 루브릭

CT 세부요소	기초 (Basic)	발달 (Developing)	능숙 (Proficient)
절차 및 알고리즘	프로그램의 절차 및 알고리즘의 효율성에 개선할 부분이 많이 보임.	프로그램의 절차 및 알고리즘의 효율성에 개선할 부분이 다소 보임.	프로그램의 절차 및 알고리즘이 효율적으로 설계됨.
병행화 및 동기화	동시적 처리나 동기화 처리가 이루어지지 않거나 적절하지 않음.	동시적 처리는 적절히 이루어졌으나 객체간의 메시지 교환 처리가 없거나 적절히 이루어지지 않음.	동시적 처리가 적절히 이루어지고 객체간의 메시지 교환을 통해 프로그램 흐름의 동기화가 적절히 이루어짐.
자료 표현	변수나 리스트가 전혀 사용되지 않음.	변수 또는 리스트를 적절히 활용함.	변수와 리스트를 적절히 활용함.
추상화	사용자 자신이 만든 블록으로 구현된 것이 없음.	전체 프로그램에서 분해된 논리적 단위가 부분적으로 새 블록으로 구현됨.	전체 프로그램에서 분해된 모든 논리적 단위가 새 블록으로 구현됨.
문제 분해	알고리즘의 논리적 부분이 혼재되어 처리됨.	알고리즘의 일부를 논리적인 단위로 나누어 처리함.	알고리즘의 주요 부분 모두를 논리적인 단위로 나누어 처리함.
시뮬레이션	새 블록을 만들어 사용하였으나 파라미터는 활용하지 않음.	새 블록에 파라미터를 사용하여 파라미터 값에 따라 다른 결과를 도출하는 과정에서 부분적으로 부정확함이 보임.	새 블록에 파라미터를 사용하여 파라미터 값에 따라 다른 결과를 도출하여 적절히 활용함.

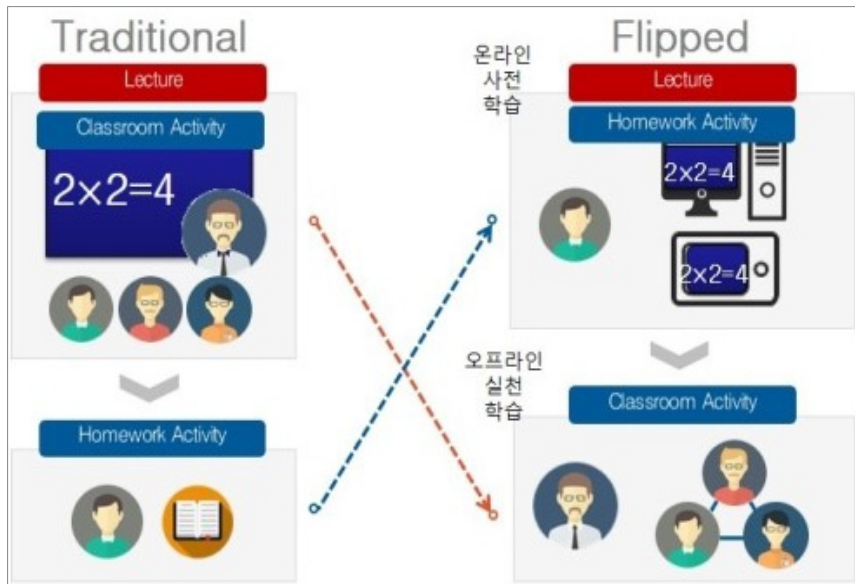
라. 플립 러닝(Flipped Learning)

최근 '플립러닝' 교육 모델에 대하여 교사 및 연구자들의 관심이 늘어나고 있으며, 스마트 기기 등의 보급과 활용이 증대되면서 교수-학습에 적용되는 사례가 증가하고 있다. 정순원 등(2013)은 플립러닝을 “기존의 교육 방식을 뒤집어, 집에서 교사가 제작한 강의를 듣고, 학교에서는 교사 및 학생들과 토론하면서 퀴즈, 프로젝트 활동, 토론 등을 통해 과제를 해결하는 교육 방식”으로 정의하고 있으며, 이종연 등(2014)은 학생이 수업시간에 배울 주요 내용을 인터넷이 가능한 집이나 개인적인 공간에서 동영상 등을 통해 자율적으로 학습하고 수업시간에는 선생님 또는 동료 학생과 함께 토론, 실습, 실험 등의 다양한 활동을 하는 것이라고 정의한 바 있다.

여러 학자들이 말한 내용을 토대로 플립러닝의 개념을 정의하면 다음과 같다. 플립러닝이란 “학생은 가정이나 일상생활에서 학교에서 배울 핵심 개념을 광대역통신망이 구축된 가정의 컴퓨터나 개인 스마트폰으로 동영상을 통해 자율적으로 학습하고 온라인으로 의사소통하며, 학교에는 선생님 또는 동료 학생들과 퀴즈, 프로젝트 활동, 토론 등을 통한 학습자중심의 교육 방식”이다. 이 교육방법은 교사가 지식전달자가 아닌 촉진자로서의 역할로 변화하게 하며, 학생들이 학교에서 프로젝트 활동 및 동료들과의 토의, 토론 등의 동료학습을 통해 배움이 일어난다는 철학에 기초하고 있으며, 새로운 교수학습 방법이라기보다 완전학습(mastery learning), 개별화 교수 체제(personalized system of instruction, PSI) 등으로 제시되어 온 개별학습의 하나라고 볼 수 있다.

플립러닝의 방법은 SW교육에 적절하게 활용될 수 있다. SW교육은 학생들에게 단순한 지식을 전달하는 것에 그치지 않고 기본적인 절차적 지식을 바탕으로 서로 협업하여 주어진 과제(문제)를 해결할 수 있도록 하는 과정이다. SW교육은 수업 대부분의 시간을 학생들 스스로 생각하며 배울 수 있는 수업, 학생들의 적극적인 참여를 유도하는 수업, 학생과 교사, 학생과 학생이 상호작용하며 협력하는 수업을 통해 학생들의 자발적인 학습이 일어나게 하는 것에 핵심

을 두고 있다. 이처럼 플립러닝을 소프트웨어 수업에 적용한다면 학습자는 사전 학습을 통해 자신의 속도로 배우고 익힐 수 있고, 스스로 학습의 속도를 선택할 수 있으며, 실제 수업시간은 학생들의 적극적인 상호작용과 협력학습을 용이하게 해주어 교실수업은 학생이 참여하는 다양한 학습으로 능동적이고 즐거운 배움이 가능할 수 있다.



[그림 II-12] 플립 러닝의 개념도⁸⁾

Salman Kahn의 칸 아카데미나 OCW, MOOC 등이 세계적으로 성공을 거두고, 여러 실증 사례들이 보고되면서 플립 러닝 학습법에 대한 관심이 높아졌으며, 국내에서는 Kahn(2012)의 책이 출판되고, 플립 러닝 교수학습을 적용 사례가 ‘21세기 교육혁명-미래교실을 찾아서⁹⁾’라는 제목으로 방영되면서 관심이 고조된 바 있다.

8) 출처 : 한국교육학술정보원 (2015). 2015년 초등 소프트웨어 교육 선도교원 연수교재. in press.
 9) KBS 파노라마-21세기 교육혁명: 미래교실을 찾아서, 제1부 거꾸로 교실의 마법(2014. 3. 20), 제2부 가르침시대의 종말(2014. 4. 3), 제3부 진짜세상을 향한 교실(2014. 5. 30).

플립 러닝은 다양한 수업 방법이 복합적으로 사용될 수 있기에 구조화된 모델로 제시될 수는 없다. 그러나 구체적으로 제시된 개념들은 실제 수업을 설계하는 데 사고의 방향성을 정립하고 하나의 가이드라인으로 작용할 수 있다는 점에서 2가지 선행 연구를 살펴보았다. 하나는 텍사스 대학 교수학습센터에서 제시한 이전의 수업과 비교한 수업 구조 및 활동내용(University of Texas at Austin Center for Teaching and Learning, n. d.: 이지연, 김영환, 김영배 2014에서 재인용)이며, 다른 하나는 이동엽(2013)이 제안한 ADDIE 모델 기반 플립 러닝 교수학습 설계 모델을 차용하여 본 연구에 제시하였다.

〈표 II -25〉 기존 수업과 플립 러닝 적용 수업 구조 및 활동내용 비교
(이지연, 김영환, 김영배, 2014)

구분	기존 수업	플립 러닝 적용 수업
수업 전	<ul style="list-style-type: none"> · 학습자에게 읽어야 할 과제 배정 · 교수자는 강의를 준비 	<ul style="list-style-type: none"> · 학습자는 제공된 모듈에 따라 학습을 하고 내용관련 질문은 기록해 둠 · 교수자는 여러 가지 학습내용을 준비
수업 도입	<ul style="list-style-type: none"> · 교수자는 무엇이 도움이 될 것인지에 대한 일반적인 가정을 함 · 학습자들은 기대학습에 대한 제한된 정보 보를 가짐 	<ul style="list-style-type: none"> · 학습자들은 (교수자에게) 그들 학습을 안내해 줄 특별한 질문을 함 · 교수자는 학생들이 가장 도움을 필요로 하는 곳을 예상할 수 있음
수업 중	<ul style="list-style-type: none"> · 교수자는 모든 학습자료를 사용하려고 노력함 · 학습자들은 수업을 따라가려고 함 	<ul style="list-style-type: none"> · 교수자는 피드백과 소규모 강의를 통해 학습자들의 과정을 안내함 · 학습자들은 배워야할 기능(skills)을 실습함
수업 후	<ul style="list-style-type: none"> · 학습자들은 보통 지연된 피드백을 받으며 숙제를 하려고 함 · 교수자는 지난 과제에 대해 평가함 	<ul style="list-style-type: none"> · 학습자들은 교수자의 명확한 설명과 피드백에 따라 그들의 지식과 기술을 계속 활용함 · 교수자는 필요한 것은 어떠한 것이라도 추가적인 설명과 리소스를 제시하고 질 높은 작업에 대해 점수를 부여함
일과 시간 중	<ul style="list-style-type: none"> · 학습자들은 공부한 것에 대한 확인을 원함 · 교수자는 종종 수업시간에 있었던 일을 반복함 	<ul style="list-style-type: none"> · 학생들은 그들이 필요로 하는 것이 어디에 있는지 도움을 구하기 위한 능력을 갖추 · 교수자는 학생들이 더 깊은 이해를 하도록 지속적으로 안내함

<표 II-26> ADDIE 기반 플립 러닝 교수학습 설계 모델(이동엽, 2013을 재구성)

단계	구성요소
	<p>교사와 학생 특성</p> <ul style="list-style-type: none"> · 수업을 위해서 교사와 학생이 온라인 학습에 대한 정보를 찾거나, 개별적으로 집에서 학습할 수 있는 능력이 있는지의 여부 등을 분석
<p>수업 요소 분석</p>	<p>수업목표 및 내용 특성</p> <ul style="list-style-type: none"> · 해당 수업을 통해 달성하고자 하는 수업목표와 다루고자 하는 내용의 수준과 범위를 고려하여 학생들이 미리 학습해야 하는 내용에 대한 계획을 세울 수 있도록 준비 <p>수업환경 특성</p> <ul style="list-style-type: none"> · 학생들의 정보화 환경, 교실의 정보화 환경 및 교실 유형에 대해 분석 · 학생들이 집에서 선행학습을 할 수 있는지 여부를 결정 · 교실에서는 어떠한 방식으로 수업을 진행할 것인지 결정
<p>수업 내용 선정 및 재구성</p>	<ul style="list-style-type: none"> · 수업목표 달성을 위한 수업 내용, 교수학습 활동 및 자료 활용 전략 수립 · 학생들이 선행학습을 해야 하는 내용 결정 · 교실 수업에서 다룰 내용 결정
<p>선행학습을 위한 교수학습 활동 선정</p>	<ul style="list-style-type: none"> · 학생들이 선행학습을 해야 할 내용에 대한 접근 가능 여부를 파악 · 필요할 경우 교사가 직접 선행학습용 자료를 제공
<p>수업 설계</p>	<p>교실수업을 위한 교수학습 활동 선정</p> <ul style="list-style-type: none"> · 선행학습을 통한 학습자의 학습 수준을 고려 · 개별화 학습 진행 시 학습한 지식에 대한 재구조화나 혹은 그러한 지식을 통해 심화학습이 일어날 수 있도록 수업을 설계
<p>효율적 플립 러닝을 위한 전략 수립</p>	<ul style="list-style-type: none"> · 학생들의 효율적인 선행학습 수행 및 실제 교실 수업 진행에 대한 전반적인 전략을 고려 · 실제 교실 수업 도입-전개-정리 단계별 교수학습 활동 · 학습 내용에 대한 보충자료 제공 방법 및 심화학습을 위한 수업 방법 설계, 교사와 학생의 상호작용 및 학생들 간의 상호작용 등 다양한 교수학습 방법에 대한 설계

<표 II -26> ADDIE 기반 플립 러닝 교수학습 설계 모델(계속)

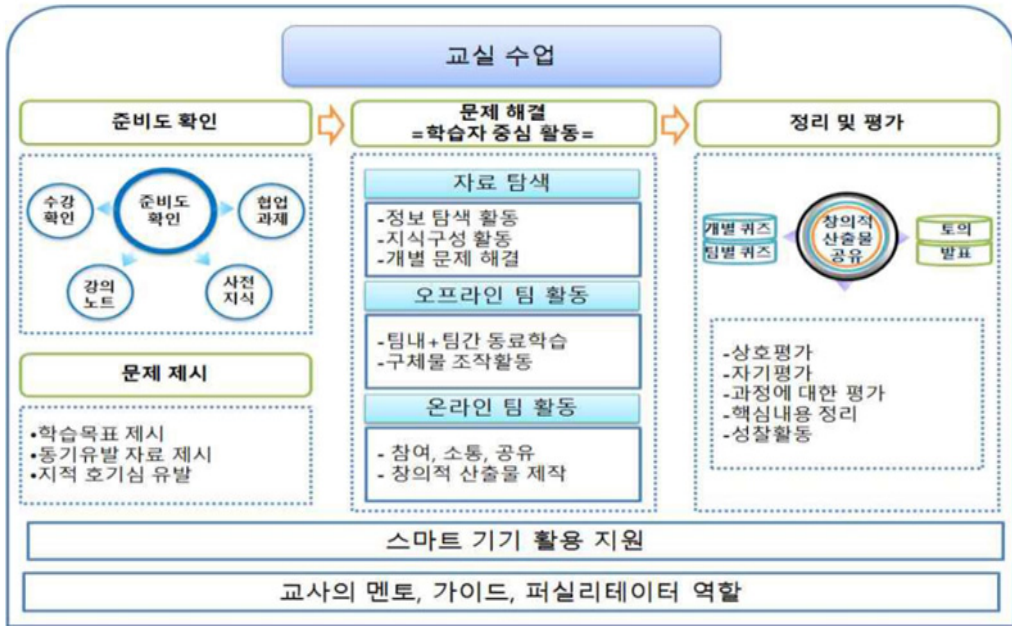
단계	구성요소
	<p>선행학습 자료 개발</p> <ul style="list-style-type: none"> · 필요 시 선행학습 자료 및 동영상 자료 개발 · 수업 목표 달성을 위한 효율적인 자료의 제공 및 교사가 학생들과 공유하고자 하는 자료의 제공
수업 개발	<p>교실수업 수업방법 및 자료 개발</p> <ul style="list-style-type: none"> · 수업 목표에 따라 교실수업의 형태를 다양하게 변형하고, 각 단계별 수업 방법 및 자료를 개발하는 것 · 학생들의 선행학습 수준에 따라 토론식 수업, 팀 구성 문제기반학습 등 다양한 형태로 교실수업을 진행
	<p>수업 단계별 활동 전략 수립</p> <ul style="list-style-type: none"> · 선행학습 및 교실수업에서 필요한 각 단계별 활동 전략 수립 · 교수학습 전략에 대한 단계별 활동 전략 수립
	<p>수업 준비</p> <ul style="list-style-type: none"> · 수업에 필요한 자료와 도구, 매체의 준비
수업 실행	<p>수업 실행</p> <ul style="list-style-type: none"> · 수업 실행 전략의 수행 · 학생들의 선행학습 여부 확인 · 교사-학생, 학생-학생, 학생-교수학습 자료 간 의도했던 상호작용의 발생 여부 확인 및 적절한 대응(피드백 제공 등) · 강의식 수업과 다른 수업 양상 등 교실환경에 대한 조치와 대응
수업 평가 및 성찰	<p>실행 과정 평가 및 성찰</p> <ul style="list-style-type: none"> · 교사가 계획했던 수업 전략 및 수업자료, 도구 및 매체의 적절한 활용 여부 · 학생의 흥미와 집중도, 개별화 학습(보충 또는 심화) 발생 여부
	<p>실행 결과 평가 및 성찰</p> <ul style="list-style-type: none"> · 수업 결과에 대한 평가를 통해 실행한 수업에 대한 반성 및 향후 수업을 위한 개선점 도출
	<p>플립 러닝 설계모델 반성 및 성찰</p> <ul style="list-style-type: none"> · 의도한 목표의 달성 여부 · 교수학습 설계의 문제가 있는 단계 분석을 통한 개선점 도출

지금까지 살펴본 플립 러닝의 성공이 수업 전에 학습을 지원할 수 있는 IT 기술의 발달에 기인한다는 점, 개별화된 학습을 통해 교사의 역할과 학생의 역할을 새롭게 제시하고 있다는 점에서, 학습자가 새로운 문제 상황에서 핵심 개념과 원리를 적용하여 해결해 볼 풍부한 기회를 제공하고, 결과보다 과정 중심의 평가를 꾀하고 있는 SW교육에 한 가지 새로운 교수-학습 방법으로 제안될 수 있으리라 여겨진다.

사전학습의 단계에서는 학생들이 가정에서 미리 SW교육에서 필요한 기본 개념을 동영상을 보고 학습을 할 수 있다. 학습자들이 수업에 대한 기본 개념에 대한 동영상강의를 미리 듣고 오는데 단순히 동영상으로 시청하는 것이 아닌 상호간에 생각을 공유 할 수 있는 장을 마련해야 한다. 이러한 장은 교육용 LMS를 통해 학습자들 사이의 협동을 지원하는 도구로 활용할 수 있다. 동영상 제작 시 주의할 점은 소프트웨어의 개념과 관련된 컴퓨터 과학의 객관적 지식과 원리로 구성한다. SW교육 주제 당 5분 단위의 섹터로 나누어 소단원에 3개 내외의 비디오 강의를 만들고 수시로 볼 수 있도록 제공하며 동영상을 본 후 퀴즈, 마인드맵, 댓글 토론, 포스트잇의 형태로 학습의 정도를 확인하는 과정이 필요하다. WSQ전략¹⁰⁾을 활용하는 것이 일반적이다.

SW교육 동영상 강의를 본인의 학습수준에 알맞게 조절하여 들을 수 있도록 하여야 하며 동영상 강의를 들었는지 교사의 확인이 필요하다. 만일 SW 동영상을 미리 보지 못한 학생들이 있다면 아침자습시간이나 쉬는 시간을 이용하여 간단히 영상을 확인하는 절차가 필요하다.

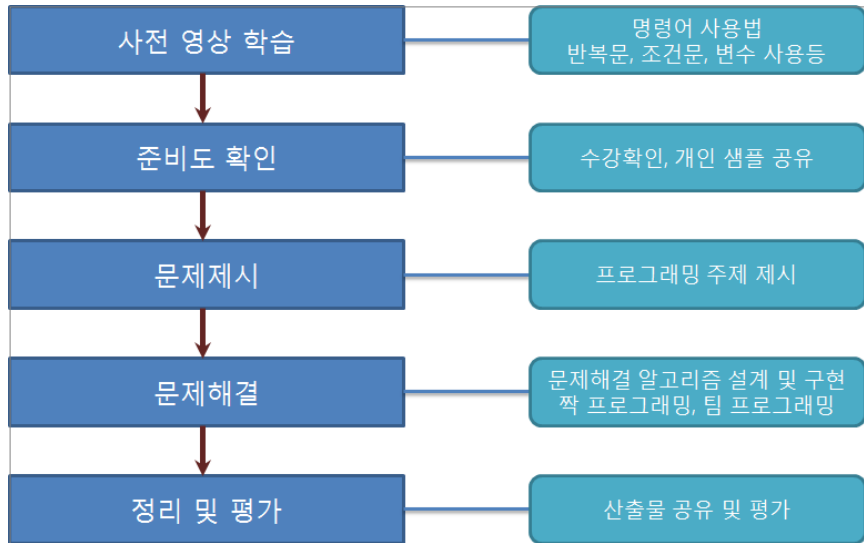
10) Raths(2013)가 “9 Video Tips for a Better Flipped Classroom”에서 제시한 WSQ 전략을 설명하면 다음과 같다. “WSQ”는 매우 간단한 아이디어이지만 대부분의 아이디어처럼 그것을 사용하는 방법에 따라 교사들이 적용하는 다양한 전략으로 활용할 수 있다. 교사들이 적용하는 방법을 선택하고 수업에 가장 적합하게 활용하도록 개선할 수 있다. WSQ는 시청하고(Watch), 요약하고(Summarize), 질문(Questions)하는 상호작용 전략을 의미한다.



[그림 II-13] 스마트교육 기반 플립러닝 수업모델(김상홍, 2015)

보다 구체적으로 SW교육 수업에서의 플립러닝은 김상홍(2015)의 제안을 수정하여 아래의 [그림 II-14]와 같은 절차로 단순화하여 적용할 수 있을 것이다. SW교육에서는 EPL이나 프로그래밍 언어에 대한 학습은 필수적이지만 한정된 수업 시간에 학생들에게 프로그래밍 언어에 대한 문법이나 사용법을 모두 가르칠 수 없으므로, 이런 부분을 플립러닝 수업 방법으로 해결할 수 있다.

또한 실제 수업에서 해결해야 하는 문제 혹은 주제가 기존 모듈을 확장하거나 기존 프로그램을 수정 보완해야 하는 형태라면 기존의 간단한 기초 예제를 미리 동영상 학습을 통해 내용을 숙지해 오게 하고, 실제 수업에서는 이를 발전시켜 문제를 해결하게 하는 형태의 플립러닝 수업이 가능하다. 사전 학습영상은 수업에 필요한 기본적인 개념의 내용으로 구성할 수 있다.



[그림 II-14] SW교육을 위한 플립러닝의 수업 흐름

Ⅲ. 컴퓨팅 사고력 신장을 위한 교수-학습 모델 개발

1. 교수학습 모델의 개발 근거

지금까지 정보 교과에 제안된 교수학습 모델은 무수히 많다. 이는 정보 교과 교수학습 모델 또는 방법이 명확하게 정해져 있다기보다 상황에 맞춰 달라질 수 있음을 시사한다(이태욱, 유인환, 이철현, 2001; 이태욱, 최현중, 2015).

본 연구에서는 수업에의 적용 가능성, 즉 교사의 활용성을 고려하여 교수학습 모델은 ‘단위 시간 내에 수업 목표를 달성하기 위해 일련의 절차와 단계를 가진 과정의 틀’로, 교수학습 전략은 ‘수업 목표를 도달하기 위해 사용하는 모델의 복합적 구성 또는 모델 내의 단계를 효과적으로 지원하기 위한 기법’으로 정의하였다.

가. 교수학습 모델의 설계 범위

1) 교육과정 재구성 수준

- 소프트웨어교육 운영지침에 의거하여 그에 맞는 내용을 수업에 적용하여 소프트웨어교육 전반 내용에 적용할 모델을 설계한다.
- 소프트웨어교육 운영지침에 제시된 내용을 재구성하여 적용하되 프로젝트 중심의 전개 또는 일부의 내용에 집중하여 알고리즘, 프로그래밍, 문제해결에 대한 기능과 관련된 내용을 중심으로 설계한다.

본 연구에서 교육과정의 재구성이라 함은 내용 재구성, 차시 재구성, 교과 간 연계 재구성을 의미한다. 차시의 재구성은 다음 제시된 같다.

2) 시수 수준

- 한 차시 이내의 활동 수준으로 선정하여 모델을 설계한다. 단위시간의 수업 설계는 절차중심의 모델과 전략 중심의 모델 모두 사용할 수 있다. 다음 장에 제시할 모델 중 DMM모델, UMC 모델, DDD모델이나 전통적인 문제해결학습, 직접교수학습모델, 개념형성모델 등에 적용 가능하다.
- 몇 개 차시 혹은 몇 주 간에 걸쳐 실행될 수 있는 활동 수준의 수업 모델 설계는 프로젝트형의 모델로 장기간의 수업진행에 따라 교육과정을 재구성하여 적용 가능하다. 전통적인 프로젝트 수업 모델이나 새롭게 제시된 NDIS모델을 중심으로 적용 가능하다.

나. 수업 모델 설계를 위한 이론적 근거

정보 교과는 앞서 논의한 것처럼 상황에 맞게 변화를 주도하는 교과로서 기존에 발표된 교육사조의 전체 내용을 포함하고 있다. 다양한 교육사조의 특징에 따라 추구하는 방향을 바르게 설정하고 그에 맞는 교수학습모델을 설정해야 한다.

1) 행동주의

행동주의 관점에서 정보교육은 실천적 학문으로서 학생들의 전문적인 기능과 프로그래밍과 코딩의 반복학습을 통해 소프트웨어 산출의 과정을 이해한다. 반복학습적인 교육으로 코딩의 기능을 몸으로 체득하고 이를 바탕으로 자신의 전문적 소질을 개발하고 미래 직업적 소양을 발견하도록 한다. 이때, 코딩교육 같이 프로그래밍 스킬을 신장하기 위한 직업적 소양 능력 신장이나 컴퓨터 과학의 기능 위주의 수업으로만 제한되는 것을 조심해야 한다(한선관, 2007).

2) 인지주의

인지주의 관점에서 정보교육은 인지주의의 기본 철학인 사고하는 주체로서 인간을 핵심으로 보고 뇌의 내부에서 일어나는 능동적인 사고 과정과 인지 구

조를 중시한다. 그러므로 SW교육에서 결과보다는 과정을 중시하며 인간의 뇌속에서 정보가 어떻게 처리되고 저장되고 재현되는지에 대하여 관심을 갖는다.

인주주의 특히 정보처리이론은 컴퓨터를 이용하여 정보를 처리하는 이론과 일치되는 부분이 많기 때문에 정보를 처리하는 과정에서 자연스럽게 인간을 이해하고 컴퓨터로 처리하는 근본적인 원리와 개념을 인식하게 된다.

인간에 대한 이해와 컴퓨터 처리에 따른 이해의 상호보완적 지식을 통해 고급적인 사고력 즉, 절차적 사고, 논리적 사고, 수렴적 사고를 비롯한 메타적 사고를 신장시킬 수 있다(한선관, 2007).

3) 구성주의

구성주의 관점에서 정보교육은 창의컴퓨팅의 관점으로 접근하며 알고리즘과 절차적 사고를 통한 CT기반의 문제해결력 신장의 능력과 함께 확산적인 창의적 사고와 팀 프로젝트를 통한 대인관계, 협력, 의사소통, 미래 진로에 대한 디자인 사고 등이 주요 이슈로 다뤄질 수 있다. Papert의 로고에 의해 시작된 컴퓨팅 파워 기반의 구성주의는 기존 교육을 변화시키는 교과로서 소프트웨어교육의 역할과 미래를 창조해가는 혁신적 교과의 성격과 많은 부분 부합된다.

특히 사물의 디지털화를 주도하는 메이커 운동(Maker Movement)에서 중요시하는 학생 중심의 능동적 만들기 활동과 교육에 술선수범하여 참여하는 과정을 통해 자신의 역량을 보여주는 구성주의 사조가 소프트웨어교육과 맞물려 미래교육의 거대한 변화로 나타나고 있다(한선관, 2007).

4) 교육사조에 대한 고찰

정보교육의 내용과 방법 그리고 평가의 단계에서 적절한 교육사조에서 추구하는 목표를 선정하여 교육의 방향과 교육과정을 구성해야 하며 그에 따른 거시적 교육의 목표를 추구하느냐에 따라 정보교육의 방향을 설정할 수 있다.

초중등 교육과정에서 전인적 인간 양성을 목표로 한다면 코딩교육과 프로그래밍교육에서 요구하는 직업적 전문성 신장의 효과적인 방법인 행동주의나 인

지주의 관점에서 추구하는 내용뿐만 아니라 구성주의 관점에서 삶을 살아가는 구성원으로서의 다양한 지식과 기능, 행동, 태도, 사회적 관계 등을 함께 할 수 있는 정보교육으로서의 목표를 중요하게 다뤄져야 한다.

현재 교육의 상황에서 구성주의적 관점(팀 프로젝트학습, 의사소통, 협력, 창의력, 디자인 사고 등 21세기 학습자 역량을 달성하기 위한 관점)의 정보교육을 정착시키기 위해서는 교육시수의 확보가 필수적이기 때문에 기존의 교육과정을 넘어서는 새로운 교육 과정과 교육전략이 요구된다. 그 대안으로서 각 교과별로 분리되어 있는 교육과정을 재구성하여 융합교육의 내용과 방법으로 현장교육에 적용할 수 있다. 또한 학교 정규교육과정과 학교 밖 교육 프로그램을 연계하여 운영할 수 있도록 유연하고 혁신적인 교육 방법과 시스템을 도입하는 것도 한 가지 방법이다(한선관, 2007).

다. 수업 모델 설계의 기본 방향

이 연구에서 정보(컴퓨터) 교과를 위한 교수-학습 모델은 다음과 같은 방향을 고려하여 설계되고 개발되었다.

첫째, 학습목표 영역인 지식, 기능, 태도 중 기능 영역을 중심으로 수업모델을 개발한다.

지식영역은 기존의 수업모델을 바탕으로 한 지식의 이해를 기반으로 한 다양한 수업이 가능하며, 태도 영역의 신장은 기능 영역의 수업 과정에서 동반되는 정의적인 부분을 통해 추구할 수 있기 때문이다.

둘째, 기존 수업방법에서 중요하게 다루는 영역을 발췌하여 새로운 모델을 개발한다.

예를 들면, 발견학습모델에서 여러 사례의 분석을 통한 귀납적인 개념과 원리를 발견하는 것을 핵심으로 놀이-수정-재구성의 단계를 가진 재구성중심모델을 개발하였다. 이 연구에서 제시하는 새로운 모델도 탐구, 프로젝트, 문제해결을 바탕으로 개발하였다.

셋째, 모델 개발의 기본 가정은 학습자의 수준과 연령, 경험에 맞게 새롭게 개발된 모델들 간에 일련의 절차를 갖는다.

각각의 모델은 초중등 학교급별로 구분하여 적용할 수 있으며, 학습자의 컴퓨팅교육 경험에 따라 적절하게 적용할 수 있다.

넷째, 개발된 수업모델은 컴퓨팅 사고 신장의 목표를 달성하기 위해 각 단계별 활동 내에 컴퓨팅 사고 구성요소(분해, 패턴인식, 추상화, 알고리즘, 자동화)를 포함하도록 구성하였다.

컴퓨팅사고 구성요소를 교사들이 이해하기 쉽도록 각각의 모델에 대한 수업 자료를 부록에 제시하였다.

2. 5가지 CT 신장 교수학습 모델 개발

가. 교수학습 모델 개발 개요

교수학습모델은 행동주의, 인지주의, 구성주의 교육 사조를 고르게 가지고 있으며 수업 모델 적용 시 교사의 판단에 따라 적절한 모델과 교육 사조를 연계하여 사용할 수 있다.

그리고 지식, 기능, 태도의 영역 중 SW교육의 성격에 맞게 학습자 활동과 역량 중심의 활동을 지원하기 위해 주로 기능 영역에 맞게 수업 모델이 개발되었다. 하지만 개발된 교수학습모델이 반드시 기능영역에만 국한되지 않음을 이해하여야 한다. 어떠한 기능이 완성되기 위해서는 기능과 관련된 기초 지식과 개념의 이해가 필요하고 그에 따른 가치와 태도적인 부분도 수반되기 마련이다.

기존 교수학습모델 즉, 직접교수, 발견학습, 탐구학습, 문제해결학습, 프로젝트학습 등 SW교육에 적합한 모델을 선정하여 이와 똑같은 단계를 가진 모델로 개발되지는 않았다. SW교육의 활동 중심 모델로 이끌기 위해 기존 모델에서 핵심이 되는 단계를 바탕으로 하였다. 이에 따라 수업모델의 명명도 ○○○중

심 모델로 선정하였으며 각 단계를 영어식 약자로 표기하였다. 기존 교수학습 방법과 CT 신장의 교수학습 모델을 개괄적으로 비교하여 <표 III-1>에 제시하였다.

<표 III-1> 전통적인 교수방법과 CT 신장 교수학습모델 비교

유형 기준	교육과정 영역	관련 교육방법	CT학습모델	활동기법 예시	목표
기능	알고리즘과 프로그래밍	직접교수	시연중심 D-M-M	도제식 교수학습기법	컴퓨팅사고 신장
		발견학습	재구성중심 U-M-C	페어드 프로그래밍 기법	
		탐구학습	개발중심 D-D-D	디버깅 기법 언플러그드 기법 디지로그 기법	
		프로젝트 학습	디자인중심 N-D-I-S	디자인 사고 기법 창의성 기법	
		문제해결 학습	CT요소중심 DPAA(P)	제한문제 해결 기법	
지식	알고리즘과 프로그래밍	개념형성 발견/탐구 학습모델		언플러그드 (교구, 신체활동) 스토리텔링 기법	컴퓨터과학의 지식 이해 및 개념형성
태도	생활과 소프트웨어	가치갈등 모델 협동학습 모델		모든 활동 과정에 반드시 포함	인성, 협력, 배려, 의사소통의 가치태도 함양

교육사조	행동주의-인지주의-구성주의
------	----------------

[그림 III-1]은 개발된 교수학습모델을 학교급, 학생의 경험 수준, 활동 주제자인 교사와 학생 중심으로 구분하고, Bloom의 인지수준의 위계에 따라 배치한 그림이다.

학교급/수준		교수학습 중심	Bloom's Taxonomy	CT 신장을 위한 교수학습모형		CT구성요소	예시 전략
초 ↓ 중 ↓ 고	초급 ↓ 고급	교사	기억 이해 적용 분석 평가 창의	시연중심 D-M-M	CT 요소 중심 DPAA(P)	분해 패턴인식 추상화 알고리즘 프로그래밍	페어프로그래밍 디지로그 디자인씽킹 디버깅 언플러그드 EPL 피지컬컴퓨팅 협동학습
		학생		재구성중심 U-M-C			
	개발중심 D-D-D						
	디자인중심 N-D-I-S						

[그림 III-1] CT 신장을 위한 교수학습 모델 개요도

나. 시연중심모델(DMM 모델)

1) 관련 교육방법(직접교수법)의 개요

직접교수법은 Beriner와 Rosenshine(1977)이 개발한 부진아 프로그램에서 처음 사용하였다. 직접 교수는 기본적으로 분할정복방식 즉, 전체를 부분으로 나눈 후 각 부분들을 차례대로 학습하면 전체를 이해할 수 있다고 가정한다. 이후 직접 교수는 고도로 복잡한 기능을 수행해야 하는 우주항공기나 잠수함을 조종하는 사람들을 훈련시키기 위해 적용되었다.

직접교수법은 도제식 모델에 가깝다. 교사가 대집단의 학생들에게 새로운 개념이나 기술을 설명하고, 교사의 지시에 따라 학생들이 연습하여 이해 정도를 조사하고, 교사의 지도하에 계속 연습하도록 하는 교수 형태이다.

직접교수법은 실제적인 학습을 강조한다. 학습 과제를 할당하고, 학생 각자에게 책임을 부여하여 그것을 완수하게 하는 방식으로 수업을 진행한다. 교사의 설명이나 시범은 그 자체로 가치를 갖는 것이 아니라 학생이 교사를 모델로 하여 모방을 통해 학습하게 하려는 것이다. 직접교수방법은 가르치는 측면에서

교사 중심방법이다. 하지만 활동측면에서는 교사의 활동 격려와 융통성을 적용해 다른 수업 방법에 못지않게 학습자 중심 방법이 될 수 있다.

2) CT 신장을 위한 시연중심모델

〈표 III-2〉 DMM 모델 개요

교육방법	시연중심모델		
	시연(Demonstration)	모방(Modeling)	제작(Making)
직접교수법	교사의 설명과 시범 표준모델 제시	학생 모방하기 질문과 대답	단계적, 독립적 연습 반복활동을 통한 기능 습득

직접교수법을 바탕으로 하는 시연중심모델은 프로그래밍 언어의 문법, 실습 중심의 명령어 등을 지도할 때 유용한 모델이다. 시연-모방-제작의 단계를 거치며 교사가 모델이 되는 학습활동의 시연을 거쳐 학습자들이 질문과 대답을 통한 모방 그리고 반복적으로 단계적, 독립적 연습을 통하여 제작하는 활동 중심 모델이다. 컴퓨터 과학의 핵심인 알고리즘과 프로그래밍 활동에 적합한 모델로 교사 중심의 모델이지만, 질문과 대답을 중심으로 학생들의 모방과 제작 활동에 집중할 경우 학습자 중심의 모델로 활동을 구성할 수 있다.

3) 단계별 세부내용

〈표 III-3〉 DMM 모델의 교수학습 절차

단계명	주요 방법	세부 단계 설명
Demonstration (시연)	설명 시범보이기 예시	가르치려고 하는 핵심전략과 기능을 교사가 설명하거나 시연을 통해 학생들에게 소개한다.
Modeling (모방)	따라하기 질문, 답변	교사의 시연내용을 학생들이 그대로 따라 실습한다. 실습의 과정에서 질문을 통해 학습자들이 교사들의 시연을 모방(modeling)한다.
Making (제작)	만들기 반복활동	시연과 모방의 단계에서 배운 내용을 토대로 학생이 직접 만들어보는 활동을 한다. 반복적으로 진행하되 단계적, 전체적인 활동을 학습자들이 전개한다.

○ 시연(Demonstration)

- 시연 단계에서 사용될 자료는 컴퓨팅 사고를 기반으로 설계되어야 한다.
- 기본적인 프로그래밍 언어나 문법을 설명하고 사용방법에 사용되는 주제라 할지라도 컴퓨팅 사고의 관점에서 접근하도록 준비한다.
- 시연 내용의 안내도 교사가 의도적으로 컴퓨팅 사고의 관점으로 전개 하되 어려운 용어(분해, 패턴, 추상화 등)는 사용하지 않고 그 의미를 내포하여 학습자들이 자연스럽게 인식하도록 시연한다.
- 특히 교사가 일련의 모듈을 시연할 때 문제의 단위를 분해하거나 알고리즘의 단계를 분해하여 그 패턴을 이해하고 핵심 개념이나 원리, 공식 등을 소개하며 컴퓨팅 구성요소를 시연한다.

○ 모방(Modeling)

- 학습자들의 모방은 교사의 컴퓨팅 사고 과정을 모델링하는 과정으로 간단한 문법의 사용부터 알고리즘의 모듈을 따라하며 컴퓨팅 사고의 구성요소를 이해한다.
- 모방의 단계에서 중요한 점은 질문과 대답을 통한 활동 지식의 이해이다. 교사의 발문에서 자연스럽게 컴퓨팅 사고를 이끌어 내는 노력이 필요하다.
- 모방을 통해 자연스럽게 문제의 분해, 변수의 설정, 발견 및 탐구를 통한 추상화의 단계를 이해하게 된다.

○ 제작(Making)

- 제작은 학생 주도적 활동을 이끄는 단계로 SW개발에 따른 제작 역량을 키운다.
- 분해된 모듈의 반복활동을 통해 패턴을 인식하고 패턴에서 발견된 개념을 추상화하도록 교사가 학습 촉진자의 역할을 해야 한다.
- 제작의 과정에서 알고리즘을 통한 절차적 사고와 프로그래밍을 통한 자동화 능력을 이끄는 과정에 집중한다.

다. 재구성중심모델(UMC 모델)

1) 관련 교육방법(발견학습법)의 개요

발견학습에서 강조되는 것은 문제해결의 기술과 학습방법의 학습이다. 교사의 역할은 학생들이 스스로 학습할 수 있도록 여건을 형성해 주는 것이다. 발견학습은 학생들이 기존의 원리나 법칙을 발견하고, 이를 통해 새로운 정보를 찾고자 하거나 새로운 결론에 도달하기 위하여 정보를 탐구, 조작, 변환하는데 적용한다. 발견학습에서는 일련의 행위나 원리를 정교화 된 하나의 알고리즘으로 다루고 있으며 이것은 가공된 형태로 학습자에게 제시된다. 학생들은 이렇게 제시된 단서를 토대로 전체의 원리나 법칙 등을 발견한다. 탐구학습과는 다른 점은 발견학습이 귀납적 접근을 하는 반면에 탐구학습은 연역적 접근을 통해 수업이 진행된다는 것이다. 발견법은 적극적인 학생이 성공의 즐거움을 맛볼 수 있어 자아실현을 촉진하고, 자주적 학습능력 · 창의적 사고력 · 귀납적 사고력 등을 자극하여 최근에 주장되고 있는 고급사고력을 함양하는 장점을 가지고 있다. 그러나 시행착오 때문에 시간이 많이 걸리고, 체계적으로 학습하지 못할 경우도 나올 수 있는 단점이 있다. 발견학습은 탐구적 학습 전략이며, 지식의 구조와 학생의 심리적 구조를 일치시킨다. 이러한 발견학습은 탐구학습과 유사하지만, 탐구학습은 연역적 접근을 하는데 비해 발견학습은 귀납적 접근을 한다. 또한, 학생들의 사고 과정을 중요시 하므로 학습자 스스로 교과를 조직하도록 요구한다.

〈표 III-4〉 발견학습법 수업 절차

단계	문제파악	예상	검증	일반화	적용
학습 과정	○선수학습요소확인 ○학습문제의 명확화	○해결방안의 직관적 예상 ○해결결과에의 예상 ○해결방향과 순서	○해결하기 ○검증하기 ○논리적 추론을 통해 결론도출	○타당성 검토 ○발견한 원리, 법칙, 성질, 공식 등을 기호화	○개념, 원리, 법칙을 유사 과제에 응용 ○수준별 과제제시

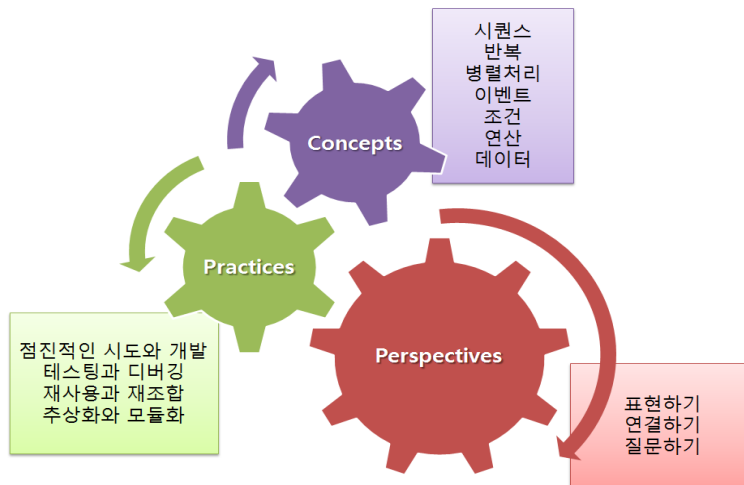
2) CT 신장을 위한 재구성중심모델

〈표 III-5〉 UMC 모델 개요

교육방법	재구성중심모델		
	놀이(Use)	수정(Modify)	재구성(reCreate)
발견학습법	학습자 체험 활동 관찰과 탐색	교사가 의도적으로 모듈 및 알고리즘을 변형하여 제시	놀이와 수정 활동을 확장하여 자신만의 프로그램 설계/제작

재구성중심모델은 발견학습법에서 사용하는 다양한 사례를 중심으로 핵심 개념과 원리를 발견하고 제시된 사례의 수정과 재구성을 통하여 컴퓨팅 사고를 이끄는 모델이다. 동기유발로 놀이를 통해 배우고자 하는 학습 모듈을 학생들이 탐색하고 사전에 준비된 모듈의 수정과정을 통하여 기능과 개념을 이해한다. 놀이 활동 및 수정 활동과 연계된 일련의 재구성 활동을 통해 학생들이 컴퓨팅 사고의 전반적인 능력을 이해하게 된다.

재구성중심모델에는 놀이를 통한 수정활동과 학습자 능동적인 재구성 활동을 통해 새로운 프로젝트를 생성하는 MIT 미디어랩에서 주로 진행했던 Creative Computing 활동의 3가지 학습요소가 나타나 있다(Brennen 외, 2014).



[그림 III-2] MIT의 Creative Computing¹⁾

3) 단계별 세부내용

〈표 III-6〉 UMC 모델의 교수학습 절차

단계명	주요 학습방법	세부 단계 설명
Use (놀이)	조작, 체험, 놀이, 활용, 탐색	학습 내용이 담긴 프로젝트를 시연해 보거나 조작해 보면서 프로젝트를 이해하는 단계이다 즉, 먼저 결과물을 가지고 놀아보며 친숙해지도록 한다. 혹은 직접 교수법을 이용하여 교사의 시범을 따라 간단한 프로젝트를 제작해가며 작동시켜 보도록 한다.
Modify (수정)	추가설계, 수정, 확장, 보완	간단히 제공된 프로젝트에 아이디어를 추가하거나 내용을 확장하여 설계한다. (새로운 스프라이트 추가 및 수정, 변수 추가, 스테이지 확장 등)
reCreate (재구성)	재구성, 구현, 개발, 산출	학습한 기능이나 내용을 활용하여 자신만의 확장된 프로그램을 설계하여 제작해 본다.

○ 놀이(Use)

- 놀이 활동에서는 다양한 사례를 사용한다. 다양한 사례라는 의미는 학생들이 즐겁게 놀이 활동을 할 수 있는 모듈로 일종의 알고리즘이나 프로그램 모듈, 완성된 소프트웨어 패키지, 피지컬 컴퓨팅 부품, 완제품을 포함한다.
- 놀이 활동의 의미는 학습자가 사용해보고 경험을 하는 활동으로 학교급이 올라가거나 경험이 많은 경우 엔터테인먼트적인 요소를 넘어 관찰하거나 탐색하기 위한 활동을 가리킨다.
- 놀이 활동에서 자연스럽게 제시된 패키지와 모듈의 일정한 패턴을 인식하고 그에 따른 놀이 절차(알고리즘) 또는 패키지 안의 알고리즘을 발견하게 한다.

11) Brennen(2014)은 창의성, 역량의 강화, 컴퓨팅에 관련된 것으로, 문제를 해결하기 위해 디자인, 컴퓨팅, 융합의 과정을 거쳐 창의적으로 문제를 해결할 수 있는 소프트웨어를 산출하는 활동으로 창의 컴퓨팅을 제안하였다. 창의 컴퓨팅의 절차 Creating, Personalizing, Sharing, Reflecting으로 제시되어 있다.

○ 수정(Modify)

- 수정 활동은 놀이 활동에 포함된 모듈 또는 패키지를 교사가 의도적으로 변형하여 제시한다.
- 학생들의 수정활동을 지원하기 위해 소스 코드의 순서 변경, 새로운 코드 채우기, 오류가 발생된 디버깅 작업 등이 사례로 제시된다.
- 수정 활동에서 프로그래밍의 문법이나 알고리즘의 이해에 관한 지식과 기능을 이해하게 된다.
- 컴퓨팅 사고의 기본이 되는 지식과 개념을 이해하도록 교사가 의도적으로 준비하되 놀이 활동에서 사용된 모듈이나 알고리즘을 변형하여 사용한다.

○ 재구성(reCreate)

- 재구성 단계에서는 새로운 소프트웨어를 개발하거나 신규 모듈을 제작하는 단계가 아니다. 앞서 놀이 활동과 수정활동에서 사용된 모듈과 패키지의 확정 버전을 만드는 활동이 주요한 내용이다.
- 이 활동 또한 교사가 사전에 준비한 자료를 바탕으로 진행하며 컴퓨팅 사고의 핵심이 되는 추상화 내용을 바탕으로 다양한 알고리즘과 자동화를 위한 프로그래밍 확장으로 자신의 지식과 기능을 구축한다.

라. 개발중심모델(DDD 모델)

1) 관련 교육방법(탐구학습법)의 개요

지식의 개념이나, 원리를 얻거나 확인하는 관찰이나, 실험 활동을 포함하는 실제적인 활동이나, 지식을 응용하여 문제를 해결하는 활동을 넓은 의미에서 보통 탐구 학습이라고 한다. 그러나 좁은 의미에서는 학생 스스로 탐구 과정을 사용하여 새로운 지식을 얻거나 문제를 해결하는 일련의 활동을 탐구 학습으로 보고 있는데 여기서 탐구는 구성주의적 관점에서 자신의 선행 지식에 기초하여 의미를 탐색하고 공유하는 과정을 포함해야 한다. 그리고 학생 자신의 지식을

실제적 활동과 문제 해결 과정에서 교사는 학생과의 상호 작용을 통해 진정한 탐구가 일어날 수 있도록 안내자가 되어야 한다.

〈표 III-7〉 탐구학습법 수업 절차

단계	문제인식 및 가설설정	탐구계획	실험	결과 해석	결론 도출
학습 내용	<ul style="list-style-type: none"> - 알아내려고 하는 것은 무엇인가? - 어떤 일이 일어날 것이라고 생각하는가? - 왜 그럴 것이라고 생각하는가? - 학생의 생각을 분명하게 하도록 도와준다. 	<ul style="list-style-type: none"> - 무엇을 어떻게 변화시키려고 하는가? - 검증을 하기 위해 무엇을 어떻게 일정하게 유지하려고 하는가? - 탐구에 필요한 사항과 절차를 확인시킨다. 	<ul style="list-style-type: none"> - 실험하는 과정에서 계획대로 했는가? - 결과를 기록했는가? - 관찰 및 측정 활동을 점검한다. 	<ul style="list-style-type: none"> - 결과를 처리하는 과정에서 자료를 적절한 표나 그래프와 같은 방식으로 나타냈는가? - 통제 변인을 고려했는가? - 해석에 필요한 사항을 준비하고 고려하도록 도와준다. 	<ul style="list-style-type: none"> - 결론을 끌어내는 과정에서 무엇을 알아냈는가? - 왜 이것이 일어난다고 생각하는가? - 어떤 규칙성이 있는가? - 결론이 자신의 원래 생각과 일치하는가?

2) CT 신장을 위한 개발중심모델

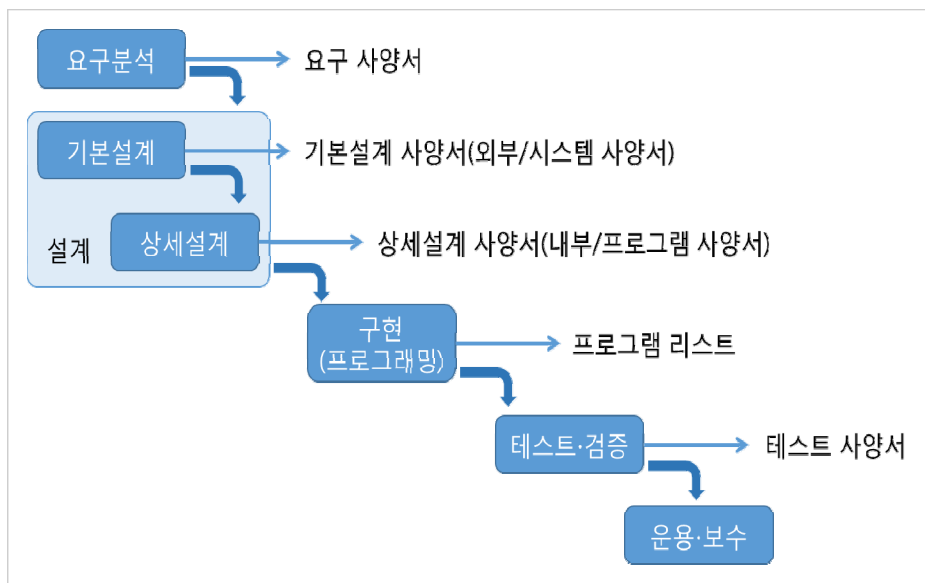
〈표 III-8〉 DDD 모델 개요

교육방법	개발중심모델		
	탐구(Discovery)	설계(Design)	개발(Development)
탐구학습법	탐색과 발견을 통한 지식 구성	알고리즘의 계획 및 설계	프로그래밍 언어로 구현 및 피드백

개발중심모델은 소프트웨어공학적인 측면에서 SW개발의 전 과정을 이해하는 모델이다. 개발하고자하는 소프트웨어에 대한 기본적인 탐구과정과 함께 개발을 위한 기초 설계의 과정을 거쳐 자신만의 소프트웨어를 개발하게 된다. 앞서 제시한 재구성중심모델과는 다르게 교사 중심의 제한된 모듈의 확장버전의 제작보다는 학습자가 개발의 과정을 주도한다. 하지만 탐구의 과정과 설계의

과정에서 교사가 의도적으로 개발의 범위를 제한하고 그에 따른 제약사항과 개발의 범위를 안내한다.

개발중심모델은 소프트웨어공학의 설계 전략을 바탕으로 한다. 소프트웨어공학의 기초가 되는 설계방법론은 ADDT(요구분석-설계-구현-시험)로 그림과 같은 폭포수 모델이 대표적인 방법이다. 폭포수 모델에서는 폭포수가 위에서 아래로 흐르듯, 요구분석, 설계, 구현, 시험, 운용 단계가 순차적으로 진행되는 소프트웨어 개발 절차이자, 소프트웨어의 생명주기를 위에서 아래로 진행하는 Top-Down 방식의 개발 모델이다. 각 공정은 작업 결과를 산출물에 모아 다음 공정으로 넘겨준다. 이 모델의 중요한 핵심은 각 단계가 명확하게 구분되어 있고, 각 단계별로 사양서를 작성하고 이를 재검토하여 문제점을 확인하고 수정한다는 점이다(Naohisa, 2011).



[그림 III-3] 소프트웨어공학설계의 폭포수 모델(Naohisa, 2011)

3) 단계별 세부 내용

〈표 III-9〉 DDD 모델의 교수학습 절차

단계명	주요 학습방법	세부 단계 설명
탐구 (Discovery)	기능의 이해, 분석, 탐색, 구현	<p>탐색과 발견을 통한 지식구성(문제분해-패턴찾기-추상화) 디버깅 문제 해결을 통한 컴퓨팅 사고 신장</p> <ul style="list-style-type: none"> ▶ 도전(챌린지) ▶ 디버깅(주어진 문제의 문제점을 발견하고 수정하면서 기능, 원리, 개념을 익힘) ▶ 컴퓨팅 활동: (재구성-진단-수정-평가)네 과정이 순환 간단한 게임을 주제로 교사가 주요 기능과 스크립트를 설명하거나 학습자들이 주어진 스크립트의 논리를 분석하여 메모한다.
설계 (Design)	알고리즘의 설계, 계획	<ul style="list-style-type: none"> ▶ 알고리즘 설계: (협력-팀워크-아이디어확정) ▶ Design Thinking: (강조-정의-창의설계- 프로토타입-공유 평가) <p>프로그래밍 언어로 구현하기 전에 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호작용 등을 이해하기 쉽게 계획한다. 또한 구현할 알고리즘을 세부적으로 생각해 보도록 유도한다. 순서도, 의사(pseudo)코드, 설계학습지 등을 활용할 수 있다.</p>
개발 (Development)	구현, 공유, 개발, 산출	<ul style="list-style-type: none"> ▶ 배운 것을 토대로 새로운 산출물 생산 ▶ 컴퓨팅과 융합: 기본 게임과 추가 설계된 내용을 프로그래밍 언어로 구현한다. <p>자신의 작품을 발표하고 피드백을 받는다. 디버깅 과정을 거치며 설계와 구현을 반복한다.</p>

○ 탐구(Discovery)

- 탐구활동에서 사용되는 주제는 교사에 의해 제시된 자료를 사용한다. 개방된 영역의 확장 주제는 다음에 제시할 디자인중심모델(NDIS)에서 다룬다.
- 탐구 활동에서 단순히 설명식 자료보다는 실제 프로그램을 분석하고 피지컬 컴퓨팅 자료의 탐색을 통하여 보다 적극적인 탐구 활동이 이루어지도록 한다.
- 탐구활동에서 문제의 영역을 분해하여 설계와 개발의 단계를 쉽게 달성할 수 있도록 제시한다.
- 개발하고자 하는 모듈과 소프트웨어의 일정한 패턴을 학생 스스로 탐구하여 추상화의 핵심 내용을 파악하도록 한다.

○ 설계(Design)

- 설계의 단계에서 사용되는 방법은 구상도, 순서도, 슈도코드 등의 전통적인 설계 방법과 함께 창의적인 공학 설계 방법을 사용한다.
- 학습자의 경험이 충분할 경우 기능 설계와 절차 설계 등의 소프트웨어 공학적인 접근 방법을 사용하여 지도할 수 있다.
- 개발을 위한 설계의 방법으로 컴퓨팅 사고의 구성요소를 단계별로 진행하며 다양한 접근 전략을 사용한다.

○ 개발(Development)

- 개발의 과정은 학습자 중심으로 진행되며 개발에 사용되는 프로그래밍 문법과 알고리즘은 교사의 준비와 조언으로 진행한다.
- 개발에 따른 전체 프로세스는 컴퓨팅 사고에 따른 분해, 패턴인식, 추상화, 알고리즘의 내용을 포함하며 그의 구현을 위한 프로그래밍 단계를 거친다. 개발의 결과는 수업시수를 고려하여 시뮬레이션이나 프로토타입 형태로 할 수 있다.
- 개발 내용에 대한 평가는 학습자들의 공유 과정과 자기반성의 관점에

서 중요하다. 개발에 대한 평가 방법은 학습자들의 프로젝트 산출물 평가와 디자인 시나리오 기법, 학습자 인터뷰를 통하여 개발의 의도, 요구분석, 설계, 개발과정, 개발의 활용에 대한 다차원적 평가가 가능하다.

마. 디자인중심모델(NDIS 모델)

1) 관련 교육방법(프로젝트학습법)의 개요

프로젝트 수업모델은 교사보다는 학생들 스스로가 문제의식을 가지고 주제를 선정하는 단계에서부터 조사나 연구, 발표 및 평가에 이르기까지 학습의 전 과정에 걸쳐 참여하는 수업 모델이다. 기존 수업 모델에 비해 교사는 안내자로서 학생들과의 활발한 피드백을 통하여 학생들이 의미 있는 학습 결과를 얻을 수 있도록 도와주는 조력자 역할을 수행해야 한다. 프로젝트 수업의 성공 여부는 교사가 얼마나 학생들이 가지고 있는 능력을 이끌어내느냐에 달려있다. 프로젝트 학습의 절차는 <표 III-10>과 같다.

<표 III-10> 프로젝트학습법 수업 절차

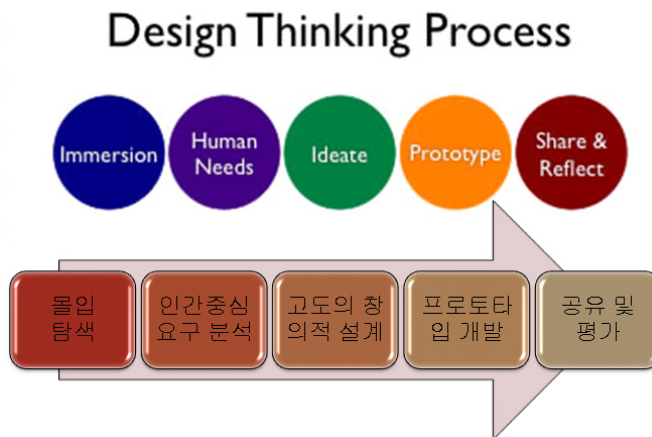
단계	주제 선정	자료 수집	계획서 제작	과제수행	결과보고서 작성	평가
내용	프로젝트의 주제를 선정한다.	프로젝트 수행을 위한 관련 자료를 수집하고 분석한다.	프로젝트 계획서를 만든다. 계획서의 내용에 대해 교사의 피드백을 받는다.	프로젝트 과제를 수행한다. 수행 결과를 교사가 피드백한다.	프로젝트 결과의 보고서를 작성한다.	프로젝트를 발표하고 그 결과에 대해 자기 성찰 공유, 평가를 진행한다.

2) CT 신장을 위한 디자인중심모델

<표 III-11> NDIS 모델 개요

교육방법	디자인중심모델			
	요구분석 (Needs)	디자인 (Design)	구현 (Implementation)	공유 (Share)
프로젝트 학습법	주어진 문제에 대한 고찰과 사용자 중심의 요구 분석	분해와 패턴찾기 알고리즘의 설계	프로그래밍과 피지컬 컴퓨팅으로 산출물 구현	산출물 공유와 피드백을 통한 자기성찰

디자인중심모델은 스탠포드 대학교의 D-school에서 제시한 디자인사고과정을 따른다. 디자인사고과정은 탐색을 통한 몰입의 과정을 통한 인간중심의 요구분석을 진행한다. SW개발이 단지 기계적인 프로그램 제품 개발이 아닌 인간의 삶을 개선하고 인류의 안전과 요구에 부합하는 활동임을 인식하여 고도의 창의적 설계를 진행한다. 컴퓨팅 사고를 신장시키기 위한 설계와 개발의 과정을 통해 프로토타입 또는 시물레이션을 제작한다. 개발된 결과를 공유와 평가를 통해 개선의 방법을 찾는 선순환 구조를 가진다.



[그림 III-4] DBL 학습과정(<http://dschool.stanford.edu/>)

3) 단계별 세부 내용

〈표 III-12〉 NDIS 모델의 교수학습 절차

단계명	학습방법	세부 단계 설명
요구분석 (Needs)	문제이해, 인간중심 요구분석	주어진 문제에 대한 고찰과 사용자에게 대한 탐색을 한다. 사용자 요구분석에서 중요하게 다루어야 할 부분은 인간중심 분석이다. 개발하고자 하는 프로그램 또는 시스템이 인간의 삶에 도움을 줄 수 있도록 안내한다.
디자인 (Design)	창의적 설계, 계획 공학적 설계	프로그래밍 언어로 구현하기 전에 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호작용 등을 이해하기 쉽게 계획한다. 또한 구현할 알고리즘을 세부적으로 생각해 보도록 유도한다. 순서도, 수도코드, 설계학습지 등을 활용할 수 있다. 이 설계과정에서 CT를 신장시키기 위한 목표를 포함하여 설계를 한다.
구현 (Implementation)	개발 및 구현 언플러그드 전략 EPL 피지컬 융합컴퓨팅	언플러그드, EPL, 피지컬 컴퓨팅의 영역을 사용하면서 학습내용을 구체화한다. 언플러그드를 통해서 먼저 관련 컴퓨터과학 지식을 알기 쉽게 이해하도록 한다. 그리고 계획한 프로젝트를 EPL을 통해 구현해 보도록 한다. 마지막으로 외부객체를 통해 디지털을 아날로그 정보로 또는 아날로그 정보를 디지털 정보로 표현해 보는 피지컬 컴퓨팅 활동을 통해 융합 활동과 연계한다. 실생활에서 사용되는 컴퓨팅의 다양한 사례를 이해하고 학생 스스로 계획하고 구현해 보는 활동을 통해 다양한 학문 또는 기술을 융합하여 표현해 보는 활동이다. 학습 내용을 실생활과 융합해 봄으로써 고차원적인 문제해결 능력을 기른다.
공유 (Share)	공유 및, 피드백	- 개발된 프로그램 공유 - 프로그램의 피드백 - 개발과정에 대한 자기성찰

○ 요구분석(Needs)

- 요구분석의 단계는 개발중심모델의 탐구단계와 비교하여 인간 중심의 관찰과 사용자 요구분석에 주안점을 둔다.
- 인간중심 요구분석은 비목적적인 소프트웨어 개발과는 달리 소프트웨어의 가치를 인간에게의 이로움과 발전적 방향을 목표로 한다.
- 개발의 과정보다는 요구분석과 디자인 과정에 더 큰 관심을 두고 진행한다.

○ 디자인(Design)

- 디자인 단계는 개발중심모델의 설계를 포함한 광의의 내용을 내포하고 있다. 단순히 설계도의 형태로 그려가는 단계를 넘어 사고의 확장과 창의적인 아이디어 산출에 집중한다.
- 디자인 단계에서 컴퓨팅 사고를 신장시키기 위한 구성요소 중 문제의 분해와 패턴을 찾는 작업을 시각화 기법을 적용할 수 있다.
- 시각화 기법은 창의성 신장을 위한 마인드맵, 브레인스토밍, 그래프와 도식화 등의 다양한 전략을 사용하고 포스트잇과 같은 생각을 손쉽게 추가, 수정, 삭제가 쉬운 도구를 사용한다.

○ 구현(Implementation)

- 구현은 앞서 제시한 모델들의 제작, 재구성, 개발 단계의 확장 내용을 담는다.
- 단순한 모듈의 개발을 넘어 통합적 시스템의 고려와 다른 영역과의 융합을 통한 실생활의 유익한 산출을 고려한다.
- 구현의 단계에서 필요한 컴퓨터 과학 지식은 언플러그드 컴퓨팅 전략으로 이해하고 프로그래밍과 피지컬 컴퓨팅의 구현을 통한 컴퓨팅 사고를 종합적으로 신장시키도록 구성한다.
- 앞선 모델과의 근본적인 차별성은 학생 중심의 활동으로 주제의 선정, 요구분석, 창의설계, 구현의 전략 등을 주도적으로 진행한다.

- 교사는 발문과 지원의 조력자 역할을 하며 구성주의 관점에서 프로젝트형 수업을 진행한다.

○ 공유(Share)

- 공유의 단계는 단순히 산출된 작품의 소개를 넘어 제작의 의도와 과정에 대한 전체 과정을 공유한다.
- 자기성찰의 단계를 통해 인간중심 요구분석과 디자인의 근본적인 평가를 하게 된다.
- 컴퓨팅 사고의 신장에 대한 교사의 발문과 조언을 통하여 컴퓨팅 파워와 추상화, 자동화의 이해를 돕는다.
- 공유 단계에서 논의된 평가결과와 의견, 자기성찰을 통하여 요구분석, 디자인, 구현의 전반적인 수정 보완 작업을 학생 스스로 결정하도록 한다.

바. CT 요소 중심모델(DPAA(P) 모델)

1) 관련 교육방법(문제해결학습법)의 개요

문제해결학습방법은 학습자 스스로 문제를 파악하고 분석하여 실행하는 모델로써, 잘못된 점을 찾아서 고쳐나가는 과정을 연습하고 결과 자체보다는 그 결과에 도달하기까지의 과정을 강조하는 모델이다. 학생의 생활에서 당면하는 현실적인 문제에서 여러 문제들을 해결해 나가는 과정을 통해 지식, 기능, 태도 등을 종합적으로 획득하여 반성적 사고를 함양하는데 그 목적이 있다.

문제해결학습의 이론적 배경은 실용주의(Pragmatism)에 기반을 두고 있다. 사고는 문제 상황에서 가장 활발히 촉진되며, 지식은 그 자체로서가 아니라 이를 활용하여 문제를 해결하는 과정에서 의미를 갖는다. 또한 Dewey(1944)의 경험주의 교육 이론을 기반으로 학습자의 지적 호기심과 흥미를 중시하고 있다. 문제 해결이란 어떤 내용을 배우는 것보다는 문제를 해결하는 방법을 배우는 것을 의미한다(Sternberg, 1996; 강숙희, 1998에서 재인용). 이러한 문제해

결모델은 여러 프로그램을 통해 교육 현장에서 활용된 바 있으며, 일반적으로 문제 확인, 해결방법 도출, 문제 해결, 일반화 네 단계의 교수학습 절차를 따른다.

〈표 III-13〉 문제해결학습법 절차

문제 확인하기	해결방법 찾기	해결하기	일반화하기
<ul style="list-style-type: none"> · 문제 상황 사례 제시 · 목표확인 · 가설 설정 	<ul style="list-style-type: none"> · 문제해결방법 탐색 · 학습 계획 및 학습 절차 확인 	<ul style="list-style-type: none"> · 문제해결 활동 · 원리 터득 · 전략 습득 여부 확인 	<ul style="list-style-type: none"> · 전략의 적용 및 연습 · 적용상의 문제점 추출, 대안 제시 · 전략의 정착 및 일반화

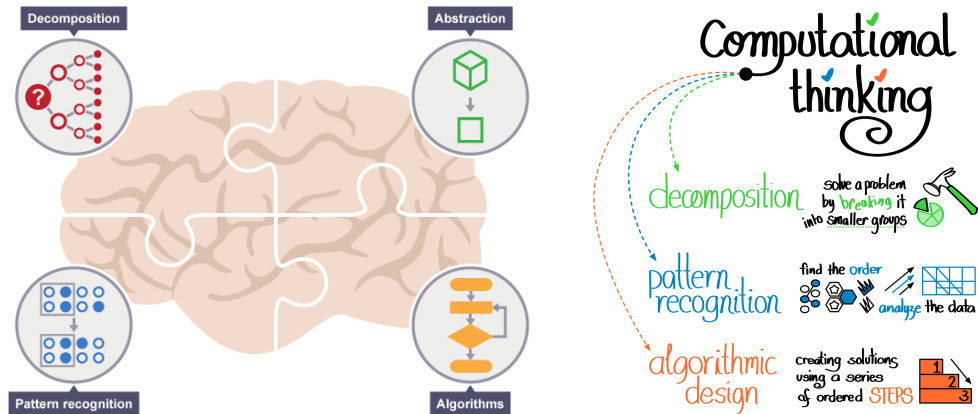
2) CT신장을 위한 CT요소중심모델

〈표 III-14〉 DPAA(P) 모델 개요

교육 방법	CT요소중심모델				
	분해 (Decomposition)	패턴인식 (Pattern Recognition)	추상화 (Abstraction)	알고리즘 (Algorithm)	프로그래밍 (Programming)
문제 해결 학습법	컴퓨터가 해결 가능한 단위로 문제 분해	반복되는 일정한 경향 및 규칙의 탐색	문제 단순화, 패턴인식으로 발견한 원리 공식화	추상화된 핵심원리를 절차적으로 구성	컴퓨터가 이해할 수 있는 언어로 구현/실행

CT요소중심모델은 본 연구의 모델에 독립적 모델로 제시하기는 어렵다. 본 연구의 목적이 컴퓨팅사고 신장을 위한 교수학습모델 개발이므로 CT요소중심 모델은 상하위계적인 측면에서 일종의 모순을 지니게 된다. 하지만 교수자들이 개발된 SW교수학습모델에서 의도적으로 CT요소를 파악하거나 제시하는 것에 어려움이 따르고 학습자들도 이러한 컴퓨팅 사고의 핵심적인 내용과 절차, 요

소를 이해하는데 더 어려움이 있으므로 개별적으로 분리하여 본 절에 CT요소 중심모델로 제시하였다. 컴퓨팅사고에 대해 국내에서 용어와 정의 그리고 실제적인 내용에 대해 의견이 분분하지만 ‘컴퓨터과학자처럼 생각하기’라는 기초적인 가정 하에 컴퓨팅 파워를 이용하여 현실에서 주어진 문제를 해결한다는 관점에서 CT요소중심모델을 고안하였다.



[그림 III-5] KS3와 구글에서 제시한 CT의 구성요소¹²⁾

CT요소중심모델은 KS3와 구글에서 제시한 4단계 모듈 전략(분해-패턴인식-추상화-알고리즘(프로그래밍))을 전제로 한다. CT요소중심모델은 일련의 단계를 가진 절차식 모델로 볼 수도 있고, 한편으로는 요소들이 분절되어 다양한 절차와 요소로 결합되는 모듈식 모델로 볼 수 있다. 즉, 분해-패턴인식-추상화-알고리즘-프로그래밍의 단계로 접근할 수 있으며, 패턴인식-추상화, 분해-알고리즘-프로그래밍, 패턴인식-추상화-분해-알고리즘-프로그래밍 등의 다양한 전략으로 지도할 수 있다. 또한 CSTA(2011)에 의해 제시된 10개 이상의 CT구성요소의 전략은 복잡하여 제외하되 필요한 경우에 자료수집, 자료분석, 시뮬레이션, 자동화 등의 다양한 전략을 추가할 수 있다.

12) 출처 : 영국 KS3 Computer Science, Google's CT.

3) 단계별 세부 내용

〈표 III-15〉 DPAA(P) 모델의 교수학습 절차

단계명	학습방법	세부 단계 설명
분해 (Decomposition)	문제의 분해	주어진 문제를 작은 단위로 분해하여 분할정복방식으로 접근한다.
	변수의 설정	작은 단위의 분해된 문제에서 사용할 변수를 설정한다.
		구현될 프로그램의 메모리와 프로세스에 대한 고민을 통하여 문제의 해결 가능성에 대한 고민을 한다.
패턴인식 (Pattern Recognition)	일정한 패턴탐색	현실세계에서 나타나는 패턴을 찾는다.
	반복항목 발견	패턴을 디지털화하기 위한 전략을 고민한다.
		프로그래밍 언어로 구현하기 위해 반복문명과 배열 메모리의 할당을 통한 기초 변수를 설정한다.
추상화 (Abstraction)	개념화	패턴인식을 통하여 발견한 공식, 원리를 설정한다.
	공식화	문제의 복잡도를 줄이기 위해 단순화한다. 개발하고자 하는 프로그램의 핵심 엔진으로 추상화의 공식(규칙, 개념 등)을 사용한다.
알고리즘 (Algorithm)	순서도	추상화된 핵심 원리를 절차화된 순서에 삽입하여 알고리즘을 완성한다.
	의사코드	알고리즘을 순서도 또는 의사 코드(pseudo code)로 표현한다. 개발된 알고리즘은 EPL, 피지컬 컴퓨팅의 영역을 사용하면서 학습 내용을 구체화한다.

한편 교육과정 개정과는 별개로 분명한 사실은 수업은 교사와 학생의 활동이 라는 점이다. 그러므로 개정 교육과정이 교실에 녹아들기 위해서는 무엇보다도 교사가 다양한 수업방법을 내면화하여 그것을 적절히 적용할 것이 요청된다. 연구를 통해 개발된 5가지 SW교육 교수학습 모델을 정리하여 제시하면 〈표 III-16〉과 같다. 또한 최종적으로 개발된 교수학습 모델을 적용한 수업 예시 자료를 개발하여 부록에 제시하였다.

〈표 III-16〉 5가지 SW교육 교수학습 모델

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

IV. 결론 및 제언

1. 요약 및 결론

2015년 문·이과 통합형 교육과정에서 적용될 소프트웨어교육의 성공적 현장 적용은 현장 교원의 교육 과정, 즉, 교육목표와 방법, 평가, 그리고 운영 편제의 이해가 전제가 될 때 가능하다. 그동안 소프트웨어교육 운영지침 개발과 교원 연수를 통해 교육과정을 이해하기 위한 기반이 마련되고 있지만, 2016년도부터 확대될 소프트웨어교육 선도학교나 연구학교에서 실제로 적용될 교수학습 방법에 대한 제시는 미흡한 실정이다. 따라서 본 연구는 SW교육의 목표로서 컴퓨팅 사고력과 구성요소, 교육과정 및 교재의 내용 분석에 근거하여 총 5가지의 교수학습 모델을 개발하고 각각의 모델에 적합한 사례를 제시함으로써 소프트웨어교육의 성공적 현장 적용을 지원하는데 그 목적이 있었다.

컴퓨팅 사고력의 개념과 구성요소에 대한 여러 논의를 살펴보면 학자에 따라 컴퓨팅 기술을 이용할 수 있는 역량에 초점을 두는 경우, 일의 처리 절차를 공식화하거나 절차적 사고에 초점을 둔 문제해결방법론의 하나로 보는 관점, 기계나 패러다임이라는 인간의 사고에 중점을 둔 경우, 광의적으로 자동화와 정보처리에 대한 연구가 확장되어 가는 과정으로 보는 등으로 다양하나, 본 연구에서는 학교 현장에서 SW교육의 안정적 정착이라는 초점을 두어 ‘컴퓨팅 사고력’이란 학생들의 일상생활에서 발생할 수 있는 문제들을 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력’이라고 정의하였다. 또한, 컴퓨팅사고력의 구성요소에 대해서도 Barr와 Stephenson(2011)과 같은 학자는 12단계로 설명하거나, CSTA는 6가지의 특성으로 규정하고 있으나, 구성요소의 각각의 요소들이 독립되어 의미를 가지는 경우도 있고, 이것이 융합되어 의미를 가지는 경우도 있으며, SW교육이 학교에 처음으로 적용되는 단계에서 구성요소가 너무 복잡하게 제시될 경우 가져올 수 있는 혼란이나

교수학습의 어려움 등을 종합적으로 고려하여 분해(D), 패턴인식(P), 추상화(A), 알고리즘(A) 등 4가지 요소로 제시하고, 프로그래밍(P)은 선택적으로 포함될 수 있다고 보았다.

이와 같은 이론적 배경을 토대로 2015 개정 교육과정이 적용되는 2018년 이전까지 SW교육 실시를 위해 마련된 소프트웨어교육 운영지침의 주요내용과 2015 개정 교육과정에서 실과, 정보과의 주요 내용, 성취기준, 시수 기준 등에 대하여 분석하고, 운영지침에 기반을 두고 학생용 교재의 내용을 분석하여 CT 신장을 위한 교수학습 모델의 실질적인 교수학습 사례를 제시하였다. 연구에서 제안된 교수학습모델은 행동주의, 인지주의, 구성주의적 관점을 고려하여 학습 목표 영역인 지식, 기능, 태도 중 기능 영역을 중심으로 수업모델을 개발하되, 기능 이외에 지식 및 태도 영역이 포함될 수 있도록 하였다. 또한 문제해결학습 등 다양한 기존 수업방법에서 빈번하게 다루어지는 모델에 착안하여 새로운 모델을 개발하고, 학습자의 특성, 학교 환경 등을 고려하여 재구성이 가능하도록 하였다. 특히 개발된 수업모델은 컴퓨팅 사고의 신장이라는 목표를 달성하기 위해 각 단계별 활동 내에 컴퓨팅 사고 구성요소(분해, 패턴인식, 추상화, 알고리즘, 프로그래밍)를 포함하도록 구성하는 등의 기본방향을 설정하고 총 5개의 모델을 개발하여 제안하였다. 즉, 컴퓨팅사고의 구성요소가 모든 학습의 목표이자 기초가 되어야 한다는 점에서 CT요소중심모델(D-P-A-A(P) 모델), 시연중심모델(D-M-M 모델), 재구성중심모델(U-M-C 모델), 개발중심모델(D-D-D 모델), 디자인중심모델(N-D-I-S 모델) 등 5가지이다. 각각의 모델에 대해서는 모델의 개요, 주요 내용, 단계별 세부 내용 등을 설명하여 각 모델이 갖는 특징과 방법에 대하여 다루었고, 실제 수업에서 어떻게 적용될 수 있는지에 대해서는 각 모델마다 학교급별 수업 자료를 제시하였다.

2. 시사점 및 제언

본 연구는 2015년 SW교육이 우리나라에서 교육과정 고시를 통해서는 처음 도입되고, 구체적으로 2015 개정 교육과정에 따라 2018년에 초등학교에서 필수화됨에 따라 SW교육을 현장에 안정적으로 정착하기 위한 교수학습 모델을 제시하고 있다는 데 의의가 있다. 또한 SW교육이 목표로 하는 컴퓨팅 사고력의 구성요소를 단순화하여 교수학습에 이용할 수 있는 모델을 제안하고 향후 3년 동안 실제 수업에 사용될 SW교육 교재를 바탕으로 수업 사례를 제시함으로써 수업을 이끌어가는 교사들이 컴퓨팅 사고력 신장을 위한 하나의 발판으로 삼을 수 있다는 점에 본 연구의 의의가 있다.

그러나 전술한 바와 같이 SW교육이 처음으로 도입되기 때문에 컴퓨팅 사고력과 구성요소에 대한 합의점이 도출되지 않았고, 학교급 간의 교육내용에 있어 범위(Scope)와 위계(Sequence)를 어떻게 가져가야 할 것인지, 수업시수가 초등학교 17차시 이상, 중학교 34차시로 되어있어 많은 학습 내용을 다루는데 한계가 있다는 점 등 SW교육과정 운영을 위해서는 광범위한 연구와 사례 적용 등의 연구가 후속적으로 이루어질 필요가 있다.

따라서 본 연구가 다양한 국내외 선행연구를 분석하고 연구진 나름의 개념 정의와 이론적 토대에 근거한 교수학습모델을 개발하여 제안하고자 했으나 다양한 교육과정 적용과 학습환경에 일반화하는데 한계를 가진다.

또한 2015 개정 교육과정에 따른 교과서가 없는 상태에서 한시적으로 적용되는 SW교육 운영지침에 근거하였고, 지식이나 태도보다는 기능 중심의 교수학습 모델을 제시하고 있다는 점에서 현장 적용 가능성과 일반화를 위한 노력이 지속되어야 한다는 과제를 남긴다. 향후 이와 같은 한계를 보완하여 다양한 교수학습모델이 개발되어 학교 현장에 적용하여 수정·보완하는 선 순환되는 구조를 통해 좀 더 정교하고 적합성이 높은 모델이 만들어져 학교 현장에 보급되고, 컴퓨팅 사고력 측정에 적합한 평가 방법에 대한 더 많은 연구가 이루어지기를 기대해 본다.

참 고 문 헌

- 강숙희 (1998). 웹의 교육적 활용에 관한 고찰. **한국교육**, 25(1), 184-200.
- 교육부·KERIS (2015). **소프트웨어 교육 운영 지침**.
- 교육부 (2015). **초중등학교 교과 교육과정**(교육부 고시 제2015-74호).
- 교육부·미래부 (2015). **SW중심사회를 위한 인재양성 추진계획**.
- 김상홍 (2015). **스마트교육 기반 플립드러닝 수업모델 개발**. 박사학위논문. 인천: 인천대학교.
- 김수환 (2011). **Computational Literacy 교육을 통한 문제해결력 향상에 관한 연구**. 박사학위 논문. 서울: 고려대학교.
- 김수환, 한선관 (2012). Computational thinking 향상을 위한 디자인기반 학습. **정보교육학회논문지**, 16(2), 319-326.
- 김수환 (2015). Computational thinking 증진을 위한 학습자 중심의 교수학습 전략의 효과. **정보교육학회논문지**, 19(3), 323-332.
- 김영수, 홍경선 (2000). 우리나라 정보과학 교육과정 연구. **교육과정연구**, 18(2), 137-157.
- 류미영, 한선관 (2015). 초등 SW교육을 위한 CT교육 프로그램 개발. **정보교육학회논문지**, 19(1), 11-20.
- 박종호, 박수중, 김종훈 (2009). “정보기술기초” 과목의 정보 통신윤리 교육의 문제점 분석 및 트렌드 설정 연구. **한국컴퓨터교육학회 학술발표대회논문집**, 13(1), 335-346.
- 서수영, 김윤희 (2003). 정보통신기술 교육의 활성화 방안에 관한 연구. **소프트웨어 미디어연구**, 2003(2), 1-21.
- 성정숙, 김현철 (2015). 국외 컴퓨터 교육과정의 변화 분석. **컴퓨터교육학회 논문지**, 18(1), 45-54.
- SPRI (2014). **SW중심사회: 의미와 대응방향**. 이슈리포트 2014-003호. 경기: 소프트웨어정책연구소.

- 유인환, 구덕희 (2004). SW 적응력 향상을 위한 SW 교수, 학습 전략. **정보교육학회논문지**, 8(4), 501-512.
- 유중현, 김종혜 (2008). 문제 해결과정에서의 정보과학적 사고 능력에 대한 개념적 고찰. **정보창의교육논문지**, 2(2), 15-24.
- 이영준, 이은경 (2008). 정보교육의 본질과 전망. **컴퓨터교육학회 논문지**, 11(3), 1-11.
- 이종연, 박상훈, 강혜진, 박성열 (2014). Flipped learning의 의의 및 교육환경에 관한 탐색적 연구. **디지털융복합연구**, 12(9), 313-323.
- 이지연, 김영환, 김영배 (2014). 학습자 중심 플립드러닝(Flipped Learning) 수업의 적용 사례. **교육공학연구**, 30(2), 163-191.
- 이태욱, 유인환, 이철현 (2001). **ICT 교육론**. 서울: 형설출판사.
- 이태욱, 최현중 (2015). **정보교과교육론**. 서울: 한빛아카데미.
- 장용준, 한선관 (2007). 정보교육을 위한 ICT 활용 유형의 분석 및 개선안에 관한 연구. **정보교육학회논문지**, 11(1), 38-48.
- 정순원, 계보경, 김재욱 (2012). **스마트교육 글로벌 동향**. 서울: 한국교육학술정보원.
- 정영식, 유정수, 임진숙, 손유경 (2015). **소프트웨어 교육론**. 서울: 씨마스.
- 정인기 (2014). 정보 교과의 '문제 해결' 주제의 성취 목표 및 교수-학습 방법에 관한 연구. **정보교육학회논문지**, 18(2), 243-254.
- 조연순 (2006). 문제중심학습의 이론과 실제. 서울: 학지사.
- 전용주, 김태영 (2015). 창의-융합적 SW교육을 위한 CT-CPS 프레임워크 개발 기초연구. **한국컴퓨터교육학회 학술발표대회논문집**, 19(1), 37-42.
- 최숙영 (2011). 21st Century Skills와 Computational Thinking 관점에서의 "정보" 교육과정 분석. **컴퓨터교육학회 논문지**, 14(6), 19-30.
- 최형신 (2014). Computational thinking 역량 계발을 위한 수업 설계 및 평가 루브릭 개발. **정보교육학회논문지**, 18(1), 57-64.
- 한선관 (2007). **정보교육방법의 실제**. 파주: 한국학술정보.
- 한선관 (2015). **창의컴퓨팅 이슈 리포트 2015-1호**. 인천: 경인교육대학교.

- Barrows H. S. & Tamblyn R.M. (1980). *Problem-Based Learning: An Approach to Medical Education*. New York: Springer Publishing Company.
- 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.
- Berliner, D. C. & Rosenshine, B. (1977). The acquisition of knowledge in the classroom, In R. C. Anderson et al., *Schooling and the Acquisition of Knowledge*. Hillsdale, NJ: Lawrence Erlbaum Associate.
- Brennan, K. & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada.
- Brennen, K., Balch, C., & Chung, M. (2015) **창의 컴퓨팅 가이드북** [Creative computing guide book]. (한선관, 김수환, 홍정미, 김현수, 신나리, 김안나, 박정환 역). 서울: 퍼플. (원전은 2014년에 공개)
<http://scratched.gse.harvard.edu/guide/download.html/>.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- Computer Science Teachers Association (2011). *CSTA K-12 Computer Science Standards*.
<http://csta.acm.org/Curriculum/sub/K12Standards.html>.
- Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7), 13-18.
- IDEO & Riverdale (2012). *Design Thinking for Educators*.
<http://www.designthinkingforeducators.com/toolkit/>.
- Dewey, J. (2007). **민주주의와 교육 : 교육철학 개론** [Democracy and education

- : an introduction to the philosophy of education]. (이홍우 역). 서울 : 교육과학사. (원전은 1944년에 출판)
- Khan, S. (2013). **나는 공짜로 공부한다** [The one world schoolhouse]. (김희경 · 김현경 역). 서울: 알에이치 코리아. (원전은 2012년에 출판)
- Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, 50(4), 36–42.
- Martin, Roger L. (2009). *The Design of Business*. Harvard Business School Press.
- National Research Council (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. The National Academics Press. http://www.nap.edu/download.php?record_id=12840.
- Raths, D. (2013). Nine video tips for a better flipped classroom. *The Education Digest*, 79(6), 15–21.
- Savery, J. R. & Duffy, T.M. (1995). Problem-based learning: An instructional model and its constructivist framework. In B. Wilson (Ed.), *Constructivist learning environments: Case studies in instructional design* (pp. 135–148). Englewood Cliffs, NJ: Educational Technology Publications.
- Naohisa, T. & Katsuhisa, M. (2013). **소프트웨어공학** [Software engineering]. (황석영 역). 서울: 도서출판 YOUNG. (원전은 2010년에 출판)
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A*, 366, 3717–3725.
- CT for educators. <https://goo.gl/iusIII>
- Google's CT. <http://goo.gl/oG0w56>

KS3, Computer Science. <http://www.bbc.co.uk/education/topics/z7tp34j/>
The d-school. 스탠포드대학교 D-School. <http://dschool.stanford.edu/>
Tim Brown (2008). IDEO's Blog.
<http://designthinking.ideo.com/?p=49/#content>

Abstract

A Research on the Development of Teaching and Learning Models for SW Education

The purpose of this paper is to develop the teaching and learning models for software education to prepare for in the actual class of the schooling. To fulfill this purpose, this study analyzed the concepts of computational thinking(CT), guideline for software education, and 2015 revision curriculum. In this study, CT is defined as thinking which can be effectively solved problems that may occur in daily life of the students and is based on the basic concepts and principles of the computing. In addition, this study set components of the computational thinking by four factors; decomposition(D), pattern recognition(P), abstraction(A), and algorithm(A). Programming(P) can be included optionally.

This paper employed two round of expert panel meeting including one round with two elementary teachers and three secondary teachers to develop and elaborate the teaching and learning models. The models proposed by this study focused on each model which include the factors of the computational thinking. The five teaching and learning models suggested by the research are as follows; 1) demonstration-oriented model(DMM), 2) recreation-oriented model(UMC), 3) developing-oriented model(DDD), 4) design-oriented model(NDIS), 5) CT element-oriented model(DPAA(P)). This study gave an overview, and described learning objective and step-by-step details about each model. Simultaneously this paper presented example instruction materials how you deal with these models in actual class depending on the school level. These models and materials could be served as a foothold in the instruction design and practice for the teachers who lead software education to improve CT ability of students.

부 록

- 1. CT 신장 SW교육 교수학습모델을 적용한 수업 자료**
- 2. SW교육 운영지침과 2015 개정 교육과정 비교**

부록 1 CT 신장 SW교육 교수학습모형을 적용한 수업 자료

〈CT 신장 SW교육 교수학습모형을 적용한 수업 자료〉는 교사가 다양한 수업 방법을 내면화하여 활용할 때 필요한 예시로서, 시연 중심(DMM), 재구성 중심(UMC), 개발 중심(DDD), 디자인 중심(NDIS) 네 가지 모델에 대한 수업 자료를 개발하였다. CT의 구성요소는 각 모델에 포함되는 요소로 적용되기에 CT 요소 중심 모델(DPAA(P))의 예시는 개발에서 제외되었다. 다시 말해서 CT 요소 중심 모델은 일반적인 교수학습 모델과 다르게 CT 교수학습모델의 이론적 근거로서 개발된 것이라 할 수 있다.

앞서 본문에서 서술한 것처럼 CSTA(2011)에서 제안한 CT 구성요소에서 〈자료 수집, 자료 분석, 구조화〉 3단계는 실제 수업에서 교사가 사전에 준비하여 제시하는 자료로 구성될 가능성이 상당하리라는 점에서, 또한 컴퓨팅 사고력만의 독창적인 구성 요소라기보다는 일반적인 문제해결 방법론의 하나일 수 있다는 점에서 CT 증진을 위한 수업을 설계하는 과정에서 제외할 수도 있다고 보았다. 여기에 영국의 KS3 컴퓨터 과학 교과와 구글에서 제안한 분해, 패턴인식, 추상화, 알고리즘(또는 알고리즘 설계)의 네 가지 요소에 자동화(프로그래밍)을 포함하여 CT 구성 요소를 재구성하였다. 이는 교육부-KERIS(2015)의 소프트웨어교육 운영지침에서 제시하고 있는 CT 구성 요소 가운데, 컴퓨팅 사고력의 핵심이 분해, 모델링, 알고리즘으로 구성된 〈추상화〉에 있다는 가정에 기반하고 있다. 다만 본 연구에서는 〈추상화〉의 의미를 ‘패턴을 만드는 일반 원칙을 정하는 것’이라는 구글에서 제안하고 있는 정의를 적용하여 제시하였다. 또한 SW교육을 실제 수업에 적용할 때 〈자동화〉가 필수적으로 포함되어야 한다는 견해를 반영한 것이다. 향후 본 수업 자료는 에듀넷(<http://www.edunet.net>) SW교육 메뉴를 통해 공개할 예정이다.

고양이와 강아지 이야기 만들기

적용학년	3학년~6학년
과목 및 차시	창의적 체험활동, 실과 (2차시)
수업모델	Demonstration - Modeling - Making 모델
수업전략	도제식 교수학습기법
도구 및 자료	블록 기반 프로그래밍 언어
중점 CT요소	■ 분해 □ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

본 수업은 고양이와 강아지 이야기 만들기라는 주제로 교육용 프로그램 언어의 문법을 배우는 수업입니다. 수업에서 학생들은 교사의 설명에 맞춰서 명령어 블록의 기능을 배우게 됩니다. D-M-M 모델은 초급 단계에서 프로그래밍의 문법과 개발의 기초 기능을 배울 때 주로 사용합니다.

교사는 프로그래밍 언어의 명령어 블록을 따라하기 형식으로 설명해 줌으로써 학생이 프로그래밍의 문법을 배우게 됩니다. 교사가 설명하고 시범을 보이고 나서, 학생들이 따라하면서 궁금한 점을 질문하게 됩니다. 그 후 학생은 교사의 설명에 따라 자유롭게 프로젝트 활동을 하게 됩니다. 수업 시간동안 이러한 과정은 순환적으로 되풀이 됩니다.

2. 학습 목표

- 교육용 프로그래밍 언어(엔트리)의 장면, 오브젝트, 명령어를 알 수 있다.

3. 수업 단계 및 활동

Demonstration (시연)	이 단계에서는 기능을 소개하고, 이 기능이 왜 필요한지 설명한다. 학생들은 블록코딩을 분해하고, 각 패턴을 인식하게 된다.
<ul style="list-style-type: none"> - 스프라이트 삭제, 추가 시연하기 - 스프라이트 움직임 시연하기 - 배경 꾸미기 시연하기 - 시작 신호 사용 시연하기 - 소리 사용 시연하기 - 고양이 스프라이트 모양 바꾸기 시연하기 - 조건문 사용 방법 시연하기 	

고양이 오브젝트를 움직여 봅시다.




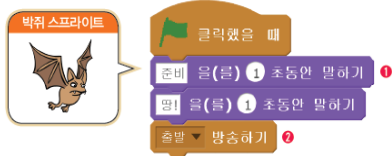

1 '이벤트' - '클릭했을 때' 블록을 끌어서 스크립트 영역에 가지고 와요.



2 제어 - '10번 반복하기' 블록을 가져와서, '클릭했을 때' 블록의 아래에 붙여요.



3 '동작' - '(10)만큼 움직이기' 블록을 가져와 '반복하기' 블록의 안에 넣어요.

<p>Modeling (모방)</p>	<p>이 단계에서는 교사의 시범에 따라하는 모델링의 과정을 통해 학생들이 하고자 하는 실행 내용에 대해 명령어로 표현하는 추상화 과정을 이해하게 된다. 아래의 예시는 방송하기의 명령의 개념과 사용방법에 대해 교사의 시범 후 학생들이 따라서 하는 장면이다.</p>
<ul style="list-style-type: none"> - 스프라이트 삭제, 추가 따라하기 - 스프라이트 움직임 따라하기 - 배경 만들기 따라하기 - 출발 신호 사용 따라하기 - 소리 사용하기 따라하기 - 고양이 스프라이트 모양 바꾸기 따라하기 - 조건문 사용 따라하기 	<p>❏ 박쥐 스프라이트가 출발 신호를 보내도록 만들어 봅시다.</p>  <ol style="list-style-type: none"> 1 '이벤트' - 'message1을(를) 받았을 때' 블록을 끌어서 스크립트 영역으로 가져와요. 2 message1을 클릭하여 새 메시지를 추가해요. 3 새로운 메시지 팝업창이 나오면 메시지 이름을 '출발'로 설정해요.  <ol style="list-style-type: none"> 1 박쥐 스프라이트를 클릭하면 '준비', '말'을 1초씩 말해요. 2 고양이와 강아지가 움직일 수 있도록 출발 신호를 보내요.  <ol style="list-style-type: none"> 1 박쥐 스프라이트에서 출발 신호를 받으면, 앞으로 10만큼 움직이는 행동을 10번 반복해요.

Making (제작) *학생 중심	이 과정에서는 학생 나름의 아이디어를 가지고 배운 명령어 문법을 사용하여 새로운 프로젝트를 개발한다. 학생들은 실제 상황에 비추어 반복연습을 할 수 있고, 실제 상황의 문제점을 탐색하고, 다른 상황에 적용할 수 있다. 이 과정을 통해 자기반성과 함께 디버깅의 작업도 할 수도 있다.
---	---

<p>프로젝트 과제 명령어 활용 미션</p> <ul style="list-style-type: none"> - 스프라이트를 추가, 삭제하기 - 스프라이트를 다양한 방향과 크기를 바꾸기 - 배경의 그림을 변경하기 - 다양한 소리를 사용하여 이야기 꾸미기 - 조건문과 반복문 사용하기 - 방송하기를 이용하여 다른 스프라이트 제어하기
<p>[프로젝트 개발을 위한 예시 자료]</p> <p>우리나라 전래동화를 사용하여 프로젝트 개발을 돕는다.</p> <p>“호랑이와 꾀감”</p> <p>“해님과 달님” 등 학생들이 많이 알고 있는 전래동화의 한 장면을 선정하여 개발하도록 안내한다.</p>

4. 평가 계획

연번	평가 기준	방법
1	스프라이트 삭제, 추가하기를 할 수 있는가?	관찰
2	배경을 알맞게 변경할 수 있는가?	관찰
3	방송하기를 효과적으로 사용할 수 있는가?	관찰
4	스프라이트 모양을 바꿀 수 있는가?	관찰
5	조건문을 상황에 맞게 사용할 수 있는가?	수행평가
6	소리를 상황에 맞게 사용할 수 있는가?	관찰

영어단어장 프로그램 만들기

적용학년	중학교 2학년
과목 및 차시	정보 (1차시)
수업모델	Demonstration - Modeling - Making 모델
수업전략	도제식 교수학습기법
도구 및 자료	블록 기반 프로그래밍 언어
중점 CT요소	□분해 □패턴인식 □추상화 ■알고리즘 ■자동화

1. 개요

교육용 프로그램 언어 중 하나인 ‘스크래치’를 이용하여, 영어 단어를 묻고 뜻을 입력하여 맞으면 정답, 아니면 오답을 말해주는 ‘영어 단어장’ 프로그램을 제작하는 과정입니다. 이 수업에 DMM 모델을 적용하기 위해서 먼저 교사가 프로젝트 구성에 필요한 리스트 구조의 개념과 리스트 블록의 사용법 및 블록 구성에 대해 설명하고, 학생들은 교사의 시연 내용을 그대로 따라 실습하고 모르는 부분을 질문을 하며 교사의 시연을 따라하도록 구성되었습니다. 배운 내용을 바탕으로 학생들이 자신만의 영어 단어장 프로그램을 만들어보고 이를 평가하는 과정을 통해 컴퓨팅 사고력을 신장할 수 있도록 설계하였습니다.

2. 학습 목표

- 변수와 리스트의 개념을 알고 활용할 수 있다.
- 자신만의 영어단어장 프로그램을 만들 수 있다.

3. 수업 단계 및 활동

Demonstration (시연)	
<ul style="list-style-type: none"> - 여러 개의 자료들을 하나로 묶는 자료 구조인 배열과 리스트의 차이점과 리스트 블록의 사용법을 설명한다. - 변수 설정과 리스트추가, 조건 구조 등 알고리즘 모듈을 설명한다. - 리스트 구조를 익힐 수 있는 단어장 프로젝트를 설명하고, 필요한 처리 과정들을 제시하여 처리 순서를 생각해보게 한 후 정리해준다. 	

알고리즘 활동	<ul style="list-style-type: none"> - 변수(위치)와 리스트(단어문제, 정답) 만들기 시연 - 리스트에 자료추가 삭제하기 시연 - 반복, 조건 구조 만들기 시연 - 단어장 프로젝트를 설명 - 필요한 처리 과정들 나열 - 순서대로 생각해보도록 한 후 정리해준다. (대답이 정답리스트의 해당 위치자료와 일치하면 '정답', 일치하지 않으면 '오답'이라고 말하기)
프로그래밍 활동	<ul style="list-style-type: none"> - 알고리즘에 해당하는 블록들을 학생들과 함께 찾아보고, 순서대로 조립하며 시연한다.

Modeling (모방)

- 학생들은 변수와 리스트를 만들고 항목들을 추가하고 삭제하는 과정을 반복하여 실습해보고 모르는 부분을 질문한다.
- 교사는 리스트 초기화를 하지 않았을 때의 문제점을 질문하고 학생들은 프로젝트를 여러 번 실행시켜봄으로써 문제점을 인식한다.
- 변수에 1씩 누적하기, 블록의 위치를 바꿔서도 실습하기 등의 추가 활동을 통해 반복구조를 이해시킨다.

알고리즘 활동	<ul style="list-style-type: none">- 변수와 리스트를 만든다.- 리스트 항목 추가, 삭제를 반복하여 실습해 본다.- 리스트 초기화 과정이 없을 때 항목들이 계속 누적되는 문제점을 알게 된다.- 반복 종료를 위한 변수 설정을 이해한다.- 조건 블록을 사용하고 반복 종료에 이르도록 위치 변수를 1씩 누적하는 실습을 해본다.
프로그래밍 활동	<ul style="list-style-type: none">- 교사의 시연을 따라 알고리즘에 해당하는 명령 블록을 조립한다.

Making (제작)

- 시연과 모방의 단계에서 배운 내용을 토대로 학생이 자신만의 단어장을 직접 만들어보는 프로젝트 활동을 한다.

알고리즘	-자신만의 단어장 만들기 프로젝트에 필요한 처리작업의 순서를 나열해본다.
프로그래밍 활동	-제시된 코드블록을 참고하여 자신의 프로젝트를 완성한다.

4. 평가 계획

연번	평가 기준	방법
1	필요한 변수와 리스트를 만들 수 있는가?	관찰
2	리스트에 항목을 추가하고 삭제할 수 있는가?	관찰
3	리스트 초기화 작업을 추가하는 위치를 말할 수 있는가?	발문
4	자신만의 단어장 프로젝트를 만들 수 있는가?	수행평가

나만의 미로 게임 만들기

적용학년	6학년
과목 및 차시	창체, 미술
수업모델	Use - Modify - reCreate 모델
수업전략	페어프로그래밍
도구 및 자료	블록 기반 프로그래밍언어
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

아이들이 접하는 소프트웨어 중에서 가장 흥미도가 높은 ‘게임’을 만들어 보는 활동으로 구성하였습니다. 초등학생의 수준을 고려하여 어려운 게임은 피하고, 간단하고 재미있게 만들 수 있는 미로 게임을 제재로 선택하였으며, 학생들은 게임의 배경이 되는 미로가 중심이 되어 장애물을 피해 목표까지 도달하는 게임을 직접 체험하고 탐색해봄으로써 게임을 구성하는 각각의 요소를 이해하고 추출하여 수정해보면서 자신만의 게임을 재구성하는 과정을 경험할 수 있도록 하였습니다.

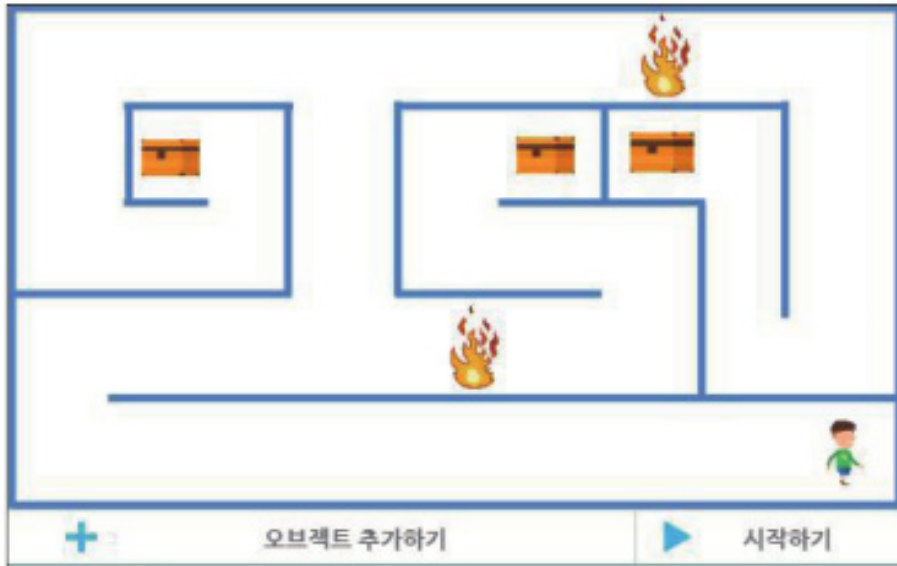
2. 학습 목표

- 간단한 미로 게임을 해보고 구성 요소를 추출할 수 있다.
- 간단한 미로 게임의 구성 요소를 바꿀 수 있다.
- 미로 게임의 구성 요소를 추가하고 재구성하여 나만의 미로 게임을 만들 수 있다.

3. 수업 단계 및 활동

Use (놀이)	
<ul style="list-style-type: none"> - 미로 게임을 해보면서 나만의 미로 게임을 만들기 위해서는 어떤 요소들이 필요할지 생각해본다. - 미로에서 장애물을 피해 마법지팡이가 들어 있는 상자를 여는 것이 목표다. - 불에 닿거나 마법지팡이가 없는 상자를 열면 상자가 쫓아온다. 	

분해 활동	<ul style="list-style-type: none"> - 게임에 필요한 요소가 무엇일지 생각해보기 - 미로 게임에 필요한 요소를 추출하기
패턴인식 활동	<ul style="list-style-type: none"> - 게임의 시작, 종료, 주인공의 움직임, 장애물의 움직임이 어떤 패턴을 갖는지 생각해보기
추상화 활동	<ul style="list-style-type: none"> - 게임의 주인공이 이동하는 방법과 벽, 장애물을 피하는 방법을 공식으로 표현하기



Modify (수정)

- 추출한 요소를 나열해보고, 미로, 주인공, 장애물 등 게임 요소에서 바꾸어서 모양이나 크기, 위치 등을 바꾸어 본다.

알고리즘 활동

- 장애물(불)이 보이는 시간을 바꾸어 본다.
- 장애물(상자)의 개수를 늘려본다.
- 장애물(스위치, 문) 등을 추가해본다.

[활동] 다음 중 하나를 골라 프로그램을 바꾸어 봅시다.

활동 1 불이 사라지는 시간을 줄이거나 늘려 봅시다.

활동 2 새로운 장애물(예: 스위치와 문)을 추가하여 봅시다.

활동 3 다음 **그림** 은 미로 게임의 재미를 높이기 위해 멀리 있는 곳을 검은색으로 가린 모습입니다. 어떤 오브젝트를 추가하고, 어떤 블록을 사용하였는지 써 봅시다.



reCreate (재구성)

- 게임을 더욱 재미있게 발전시킬 수 있는 방법은 무엇인지 생각하여 새롭게 재구성된 게임을 만든다.

알고리즘,
프로그래밍

- 새로운 장애물(스위치, 문)을 추가하기
- 장애물의 알고리즘을 바꾸어보기
- 장애물에 닿았을 때 생명력이 줄어들거나 단계를 완료했을 때 점수를 부여하기
- 미로의 모양을 바꾸거나 추가하여 새로운 단계를 프로그래밍하여 만들기

4. 평가 계획

연번	평가 기준	방법
1	간단한 미로 게임을 해보고 구성 요소를 추출할 수 있는가?	수행평가
2	간단한 미로 게임의 구성 요소를 바꿀 수 있는가?	수행평가
3	미로 게임의 구성 요소를 추가하고 재구성하여 나만의 미로 게임을 만들 수 있는가?	수행평가

장애물 피하기 게임 만들기

적용학년	3학년~6학년
과목 및 차시	창의적 체험활동, 실과 (3차시)
수업모델	Use - Modify - reCreate 모델
수업전략	디버깅
도구 및 자료	블록 기반 프로그래밍 언어
중점 CT요소	■분해 ■패턴인식 ■추상화 ■알고리즘 ■자동화

1. 개요

장애물 피하기 게임에 U-M-C 모델을 적용하기 위해서는 먼저 학습자가 장애물 피하기 게임을 실제 해 보는 과정에서 게임에 들어있는 각종 게임 요소(독수리, 장애물 등)를 파악하여 그들의 관계를 나열하고 게임 요소(독수리나 장애물의 움직임)가 가지고 있는 패턴을 파악하도록 하는 활동이 선행되어야 합니다. 다음 단계로, 교사는 의도적으로 프로그램의 일부분을 변경하여 학생들에게 제공하고 학생들은 교사와 함께 변경된 부분이 장애물 피하기 게임에 어떤 영향을 미쳤는지를 탐색해 보도록 합니다. 예를 들어, 독수리의 움직임 축을 X에서 Y로 변경하여 제공해 주는 활동을 통해 프로그램 요소들 간의 관계를 보다 명확히 파악할 수 있으며, 이동 기준축의 변경으로 인해 발생할 수 있는 여러 부가적인 예상하고 해결해 볼 수 있습니다. 마지막 단계로 학생이 중심이 되어 장애물 피하기 게임을 변경해 보는 활동을 하게 됩니다. 변경(해결)하고자 하는 여러 요소들을 학생이 스스로 결정하고 문제를 해결하는 과정을 통해 학생은 CT의 여러 요소들을 통합적으로 경험할 수 있습니다.

2. 학습 목표

- 다양한 요소를 수정하여 나만의 장애물 피하기 게임을 만들 수 있다.

3. 수업 단계 및 활동

Use (놀이)	
<ul style="list-style-type: none"> - 완성된 장애물 피하기 게임을 학생이 직접 놀아보게 한다. - 이 단계에서 학생들이 게임의 구성요소를 찾고, 게임의 구성요소들이 갖고 있는 특징과 관계, 사건 속에서 패턴을 찾아보도록 한다. 	
분해 활동	- 장애물 피하기 게임에 필요한 요소를 분해도로 표현하기 (독수리, 장애물)
패턴인식 활동	<ul style="list-style-type: none"> - 독수리의 움직이는 규칙을 찾아보기 (위, 아래로 움직인다.) - 장애물의 움직임 생각하기 (장애물은 오른쪽에서 왼쪽으로 움직인다.)
Modify (수정) - 교사중심	
<ul style="list-style-type: none"> - 교사가 의도적으로 오류를 제시하여 학생들이 수정하거나 불완전한 소스를 제공하여 학생들이 채워간다.(디버깅) - 프로그램 내에서 핵심이 되는 모듈을 학생들이 찾아낼 수 있도록 한다. 	
추상화 활동	<ul style="list-style-type: none"> - 독수리가 움직이는 방향 축을 Y축 기준에서 X축 기준으로 변경하여 제공하고 학생들이 오류를 해결하는 과정에서 핵심적인 개념 발견하기 - 움직이는 기준 축이 바뀜으로서 새롭게 발생할 수 있는 문제 예상하기
알고리즘 활동	<ul style="list-style-type: none"> - X축 블록을 Y축 블록으로 변경하기 - 독수리가 화면의 경계를 넘어가지 않도록 독수리가 이동할 수 있는 Y축 좌표값 제한하기

reCreate (재구성) - 학생중심

- 학생 스스로 변경할 수 있는 요소가 무엇인지 파악한다.
- 파악한 요소를 변경함으로써 발생하는 부수적인 문제점들을 해결한다.

분해,	- 변경하고자 하는 부분 나열하기 (요소, 환경, 규칙 등) 예) 독수리의 속도, 독수리의 크기, 장애물의 크기, 장애물의 종류 등
추상화,	- 해당 부분에 관련된 핵심 코드 찾기 예) 독수리와 장애물의 속도를 결정하는 변수 찾기
알고리즘,	예) 독수리의 크기를 변경하는 코드 찾기
프로그래밍	- 변경 의도대로 코드 수정 및 추가하기 예) 독수리 속도 변수 값 변경하기 예) 코드나 스프라이트 변경을 통해 독수리의 크기 변경하기
활동	- 테스트 및 시뮬레이션을 통해 디버깅하기

4. 평가 계획

연번	평가 기준	방법
1	장애물 피하기 게임에 필요한 요소를 말할 수 있는가?	발문
2	제시된 오류를 수정할 수 있는가?	수행평가
3	바꾸고 싶은 요소를 찾아 변경해야 할 코드를 발견할 수 있는가?	관찰
4	테스트와 디버깅을 거쳐 최종작품을 만들 수 있는가?	수행평가

복숭아 선별기

적용학년	중학교
과목 및 차시	정보 1차시
수업모델	Use - Modify - reCreate 모델
수업전략	페어프로그래밍
도구 및 자료	블록 기반 프로그래밍언어
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

본 수업은 복숭아를 무게에 따라 분류하는 프로그램을 먼저 실행해보면서 프로젝트를 이해합니다. 이후 복숭아 분류 프로그램에서 찾을 수 있는 분류 방법이나 패턴을 찾아 분석한 후 복숭아가 분류되는 조건 등을 바꿔 수정, 보완의 단계를 거쳐본 후 학습한 패턴이나 방법을 적용할 자신만의 프로젝트를 실제로 제작함으로써 U-M-C 모델을 이해할 수 있습니다.

2. 학습 목표

- 복숭아 선별기 프로그램을 체험해보고 분류 방법 패턴을 찾을 수 있다.
- 복숭아 선별 기준을 바꿔 복숭아를 분류할 수 있다.
- 복숭아를 다른 개체로 바꾸고, 선별기준을 추가하고, 재구성하여 나만의 과일 선별기를 만들 수 있다.

3. 수업 단계 및 활동

Use (놀이)

- 1) 복숭아 선별기 프로그램을 실행해 본다.
(SW교육 교재 - 중등 활동 모듈 2주제 - 복숭아 선별기)



- 2) 복숭아가 나타나고 랜덤(50 ~ 300)의 복숭아 무게가 결정된다.

최소	최대
50	300

- 3) 복숭아 크기에 따라 등급이 결정된다.

무게	등급
250g 초과	특상
200~250g 미만	상
150~200g 미만	중
100~150g 미만	하
~ 100g 미만	기타

- 4) 등급에 따라 바구니로 이동된다.
- 5) 2)~4)과정이 계속 반복된다.

Modify (수정)

- 1) 복숭아 선별기 프로그램에서 변경해 볼 수 있는 것들을 생각해본다.
- 2) 복숭아의 크기 값의 범위를 100 ~ 400으로 변경해본다.

최소	최대
100	400

- 3) 복숭아 크기에 따라 등급을 아래와 같이 변경해본다.

무게	등급
300 초과	최고여요!
250 ~ 300	아주 좋아요!
200 ~ 250 미만	좋아요!
150 ~ 200 미만	보통이에요!
~ 150 미만	문제가 있어요!

reCreate (재구성)

- 복숭아를 사과로 변경하고, 새로운 등급 규칙을 정하고 적용하여 프로그램을 완성해 봅니다.
- 새로운 규칙으로는 사과의 스크래치 유무에 따른 등급을 새롭게 추가하여 봅니다. 즉 무게가 아무리 좋아도 스크래치가 있다고 나오면 기타 (문제가 있어요!) 등급으로 이동합니다.

4. 평가 계획

연번	평가 기준	방법
1	복숭아 선별기 프로그램을 실행해보고 변경 가능한 부분을 찾을 수 있는가?	수행평가
2	복숭아의 무게나 무게에 따른 등급을 변경하여 적용할 수 있는가?	수행평가
3	복숭아를 다른 과일로 변경하고 새로운 등급 규칙을 적용하여 나만의 과일 선별기를 만들 수 있는가?	수행평가

도형 그리기 프로그램

적용학년	중학교 2학년
과목 및 차시	정보
수업모델	Use - Modify - reCreate 모델
수업전략	디버깅
도구 및 자료	블록 기반 프로그래밍 언어
중점 CT요소	<ul style="list-style-type: none"> ■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

도형 그리기 프로그램에 U-M-C 모델을 적용하기 위해서는 먼저 학습자가 완성파일을 실제 해 보면서 프로젝트의 들어있는 요소들의 관계 및 패턴을 파악하도록 합니다.

그 다음 교사는 의도적으로 프로그램의 일부분을 변경하여 학생들에게 제공하고 학생들은 교사와 함께 변경된 부분이 도형 그리기에 어떤 영향을 미쳤는지를 탐색해 보도록 합니다. 예를 들어, 연필 스프라이트의 이동거리를 증감시킨다든지 회전 각도를 변형시켜봄으로써 프로그램 요소들 간의 관계를 보다 명확히 파악할 수 있으며, 요소들의 변경으로 인해 발생할 수 있는 여러 부가적인 사항을 예상하고 해결해 볼 수 있습니다.

마지막으로 학생이 중심이 되어 도형그리기 프로그램을 변경해 보는 활동을 하게 됩니다. 변경(해결)하고자 하는 여러 요소들을 학생이 스스로 결정하고 문제를 해결하는 과정을 통해 학생은 CT의 여러 요소들을 통합적으로 경험할 수 있습니다.

2. 학습 목표

- 다양한 요소를 수정하여 나만의 도형 그리기 프로그램을 만들 수 있다.

3. 수업 단계 및 활동

Use (놀이)	
<ul style="list-style-type: none"> - 완성된 삼각형 그리기 프로그램을 실행해보도록 한다. - 이 단계에서 학생들이 삼각형 그리기에 필요한 구성요소(이동거리, 회전각도)를 찾고, 구성요소간의 관계와 패턴을 찾아보도록 한다. 	
분해 활동	- 삼각형이 그려지는데 필요한 요소를 나열해본다. (이동거리, 회전각도, 반복횟수)
패턴인식 활동	- 연필의 움직임은 규칙을 찾아보기 (100만큼 이동하고 120도 회전)
Modify (수정) - 교사 중심	
<ul style="list-style-type: none"> - 교사가 의도적으로 오류를 제시하여 학생들이 수정하거나 불완전한 소스를 제공하여 학생들이 채워간다.(디버깅) - 프로그램 내에서 핵심이 되는 모듈을 학생들이 찾아낼 수 있도록 한다. 	
추상화 활동	<ul style="list-style-type: none"> - 연필의 이동거리를 100에서 120으로, 회전 각도를 120도에서 90도로 변경하여 제공하고 학생들이 오류를 해결하는 과정에서 핵심적인 개념 발견하기 - 회전 각도가 바뀜으로서 새롭게 발생할 수 있는 문제 예상하기
알고리즘 활동	<ul style="list-style-type: none"> - 회전각도 변경하기 - 반복횟수 변경하기 - 그리고자 하는 다각형의 각의 수 증가에 따라 이동거리를 축소해야 폐다각형이 되므로 이동 거리를 500/각으로 제한한다.

reCreate (재구성) - 학생 중심
<ul style="list-style-type: none"> - 학생 스스로 변경할 수 있는 요소가 무엇인지 파악한다. - 파악한 요소를 변경함으로써 발생하는 부수적인 문제점들을 해결한다.

분해, 추상화, 알고리즘, 프로그래밍 활동	<ul style="list-style-type: none"> - 변경하고자 하는 부분 나열하기 (도형의 각의수, 도형의 크기, 도형의 개수, 도형색, 등) - 해당 부분에 관련된 핵심 코드 찾기 예) 다각형의 각의수, 반복 횟수, 이동거리 찾기 - 반복횟수, 회전수, 이동거리를 결정하는 변수 찾기 예) 도형의 모양 변경하는 코드 찾기 - 원하는 다각형의 각을 입력하면 해당 다각형이 원하는 횟수만큼 그려지도록 코드 수정 및 추가하기 예) 그려지는 도형의 개수를 변경하기 위해 필요한 블록 추가하기 - 테스트 및 시뮬레이션을 통해 디버깅하기
-------------------------------------	--

4. 평가 계획

연번	평가 기준	방법
1	도형 그리기에 필요한 요소를 말할 수 있는가?	발문
2	제시된 오류를 수정할 수 있는가?	관찰/수행평가
3	바꾸고 싶은 요소를 찾아 변경해야 할 코드를 발견할 수 있는가?	관찰
4	테스트와 디버깅을 거쳐 최종작품을 만들 수 있는가?	수행평가

로봇으로 교실을 청소하자

적용학년	6학년
과목 및 차시	실과
수업모델	Discovery - Design - Development 모델
수업전략	스토리텔링, 언플러그드, 페어 프로그래밍
도구 및 자료	언플러그드 활동지, 블록 기반 프로그래밍언어
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

본 주제는 교실 청소를 해줄 수 있는 가상의 청소 로봇의 프로그램을 만들어 보는 활동으로 구성되어 있습니다. 언플러그드 활동을 통해 실생활에서 접할 수 있는 청소 로봇의 알고리즘을 이해하고 간단하게 프로그래밍 해봄으로써 문제 해결 과정을 경험하도록 하였습니다.

2. 학습 목표

- 언플러그드 활동을 통해 청소 로봇의 동작 원리를 이해할 수 있다.
- 청소 로봇의 알고리즘을 설계하고 프로그래밍 언어를 활용하여 구현할 수 있다.
- 모양과 크기가 다른 교실을 청소할 수 있는 청소 로봇 프로그램을 만들 수 있다.

3. 수업 단계 및 활동

Discovery (탐구)

<문제 제시>

- 이야기 형태로 주어진 문제 상황을 보고 주어진 문제가 무엇인지 생각해 보게 한다.
- 해결해야 할 문제를 인식하고 문장으로 써보게 한다.



위 대화를 읽고 해결해야 할 문제를 정리하여 아래 빈칸에 채워 봅시다.

누가 청소하는 학생	언제 방과 후	어디에서 교실에서	왜 청소하기 힘들기 때문에
어떻게 알아보고,		무엇을 교실 바닥을 청소할 수 있는 을 만들어야 한다.	

[언플러그드 활동] 문제 해결 방법 탐색

분해	- 언플러그드 활동을 통해 로봇에게 명령을 내리는 방법과 로봇이 동작하는 원리를 탐색한다.
패턴인식	- 로봇이 목적지까지 가는 방법에 대한 패턴을 찾아본다. 벽을 따라간다. 지그재그로 이동한다. 가로, 세로 축을 중심으로 움직인다. 등
추상화	- 주어진 모양의 교실을 모두 청소하려면 어떤 방법을 사용할지 공식으로 표현한다. - 최단거리 알고리즘, 백트래킹 알고리즘을 통하여 청소하지 못한 공간에 대해 인식한다.

❑ 칠판 앞의 출발점에서 도착점까지 이동하는 '로봇 역할 놀이'를 해 봅시다.

역할 놀이 규칙

- 1 두 사람이 짝이 되어 로봇과 조종사 역할을 맡는다.
- 2 로봇이 앞에, 조종사가 뒤에 선다.
- 3 로봇은 조종사의 명령에 따라 움직인다.
- 4 조종사가 내릴 수 있는 명령은 "앞으로 한걸음 가", "오른쪽으로 90도만큼 돌아", "왼쪽으로 90도만큼 돌아" 세가지이다.
- 5 조종사는 한 번에 한 개의 명령만 내릴 수 있다.

- 로봇 역할 놀이를 해 보고, 청소 로봇 프로그램을 어떻게 만들어야 할지 **움직이는 방법**에서 찾아 동그라미 안에 표시해 봅시다.

움직이는 방법

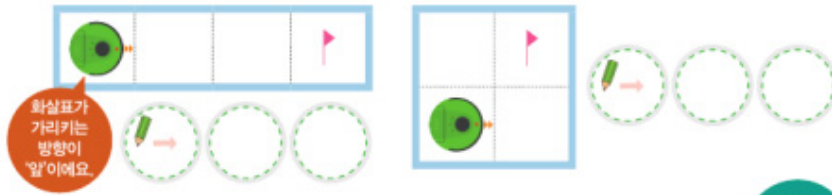
앞으로
한 칸 가기

왼쪽으로
90도만큼
돌기

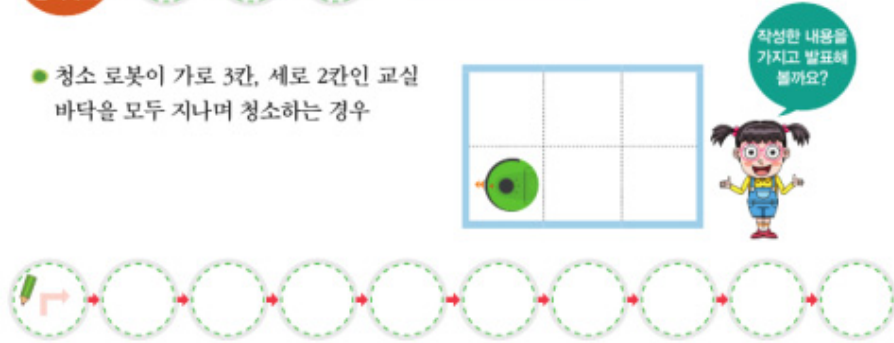
오른쪽으로
90도만큼
돌기

- TIP**
- 한 번에 이동할 수 있는 거리는 한 칸입니다.
 - 청소 로봇은 한 번에 한 가지 명령만 실행합니다.

- 청소 로봇(●)이 깃발(↑)이 있는 곳까지 청소하는 경우



- 청소 로봇이 가로 3칸, 세로 2칸인 교실 바닥을 모두 지나며 청소하는 경우



Design (설계)	
- 청소 로봇이 어떻게 이동하면 교실을 모두 청소 할 수 있는지 알고리즘을 설계하고 표현한다.	

알고리즘 활동	- 청소 로봇이 어떻게 이동하면 교실을 모두 청소 할 수 있는지 알고리즘을 설계하고 표현한다.
	- 순서대로 실행하는 알고리즘(순차)에서 반복되는 부분을 발견하여 반복에 따라 실행하는 알고리즘(반복)으로 프로그램을 설계한다.
	- 실제로 문제가 해결되는지 실행하여 보고 짝과 문제점이 없는지 서로 평가해본다.

대상	순서대로 실행하는 알고리즘
 <p>청소 로봇</p>	<ol style="list-style-type: none"> ① 오른쪽으로 90도만큼 돌기 ② 앞으로 한 칸 가기 ③ ④ ⑤ ⑥ 오른쪽으로 90도만큼 돌기 ⑦ 앞으로 한 칸 가기 ⑧ ⑨ ⑩ <div style="text-align: center; border: 1px solid black; border-radius: 15px; padding: 5px; margin: 10px 0;"> 보기 </div> <ul style="list-style-type: none"> • 오른쪽으로 90도만큼 돌기 • 앞으로 한 칸 가기
대상	반복에 따라 실행하는 알고리즘
 <p>청소 로봇</p>	<p style="text-align: center;">() 번 반복</p> <ol style="list-style-type: none"> ① 오른쪽으로 90도만큼 돌기 ② 앞으로 한 칸 가기 ③ ④ ⑤ <div style="text-align: center; border: 1px solid black; border-radius: 15px; padding: 5px; margin: 10px 0;"> 보기 </div> <ul style="list-style-type: none"> • 2 • 오른쪽으로 90도만큼 돌기 • 앞으로 한 칸 가기

Development (개발)

- 알고리즘을 바탕으로 블록 기반 프로그래밍 언어를 활용하여 청소 로봇 프로그램을 만들어 본다.

프로그래밍 활동

- 청소 로봇 프로그램을 만들어 본다.
- 다양한 활동(더 나아가기)을 통해 청소 로봇 프로그램을 발전시킨다.
- 발전된 청소 로봇 프로그램을 발표하고 피드백을 받는다.
- 설계와 구현 과정을 반복하며 더 나은 청소 로봇 프로그램으로 만들고 공유한다.







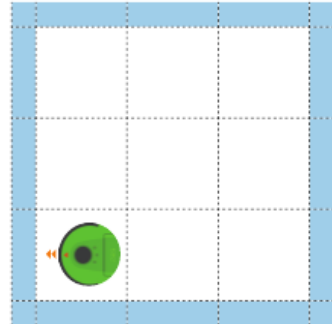
다양한 활동

다음 중에서 하나를 골라 프로그램을 바꾸어 봅시다.

활동 1 청소 로봇이 같은 바닥을 2번 이상 반복하여 청소하도록 프로그래밍해 봅시다.

활동 2 조건문을 사용하여 청소 로봇이 벽이나 장애물에 닿기 전에 방향을 바꾸도록 만들어 봅시다.

활동 3 내가 만든 청소 로봇 프로그램은 다음과 같은 모양의 교실은 청소를 하지 못하는 부분이 발생할 수 있습니다. 청소를 할 수 있도록 만들어 봅시다.



4. 평가 계획

연번	평가 기준	방법
1	언플러그드 활동을 통해 청소 로봇의 동작원리를 이해하는가?	관찰
2	청소 로봇의 알고리즘을 설계하고 프로그래밍 언어를 통해 구현할 수 있는가?	동료평가
3	만들어진 프로그램을 발전시켜 새로운 프로그램을 만들 수 있는가?	수행평가

디지털 애완동물 키우기

적용학년	중학교
과목 및 차시	정보 2차시
수업모델	D-D-D
수업전략	스토리텔링, 협동학습
도구 및 자료	블록 기반 프로그래밍언어
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

본 수업은 디지털 애완동물 키우기란 주제로 키우고자 하는 애완동물의 종류를 정하고, 애완동물의 성장에 필요한 요소들을 찾아보는 과정을 통하여 자료를 수집 및 조사하는 과정을 학습합니다.

이 후 애완동물의 성장에 필요한 요소를 충족시키는 방법을 구상하고, 충족 시켜줄 때와 그렇지 못할 때의 애완동물의 변화를 설계하는 과정을 통하여 구상, 계획, 설계 단계를 이해할 수 있습니다.

나아가 설계를 바탕으로 실제 애완동물 키우기 프로그램을 제작해 보고 다른 그룹의 작품을 보고 비교해본 후 자신들의 프로그램을 수정하여 최종 작품을 공유하는 전체 과정을 통하여 이런 일련의 과정을 통하여 CT의 요소들을 통합적으로 이해할 수 있습니다.

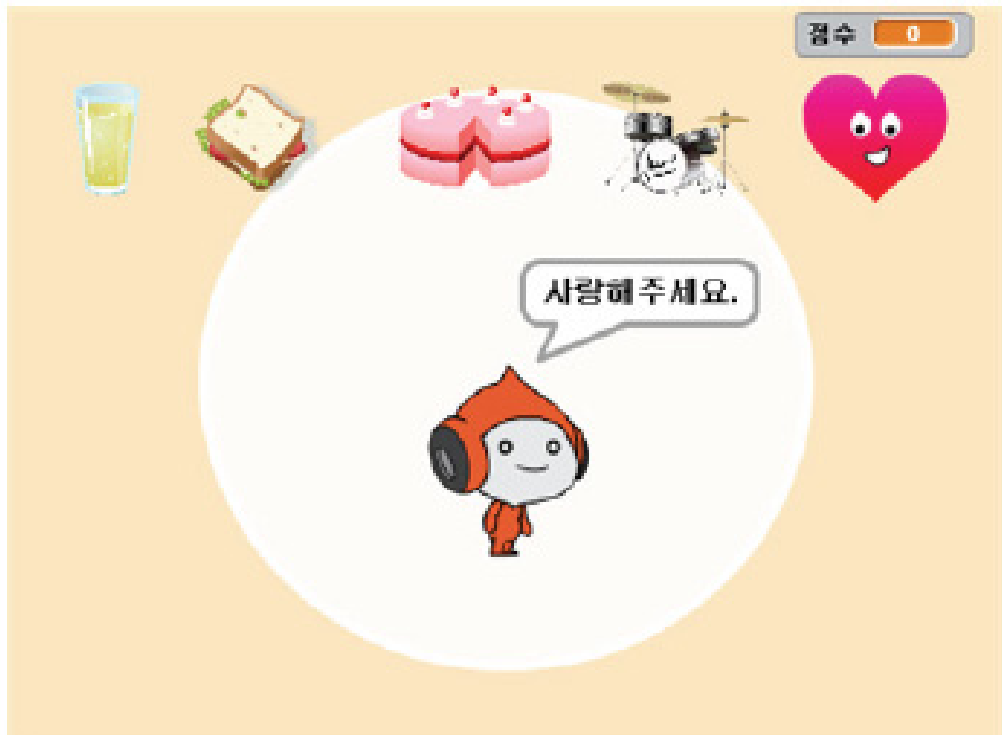
2. 학습 목표

- 디지털 애완동물 키우기 게임의 알고리즘을 설계하고 프로그래밍 언어를 활용하여 구현할 수 있다.

3. 수업 단계 및 활동

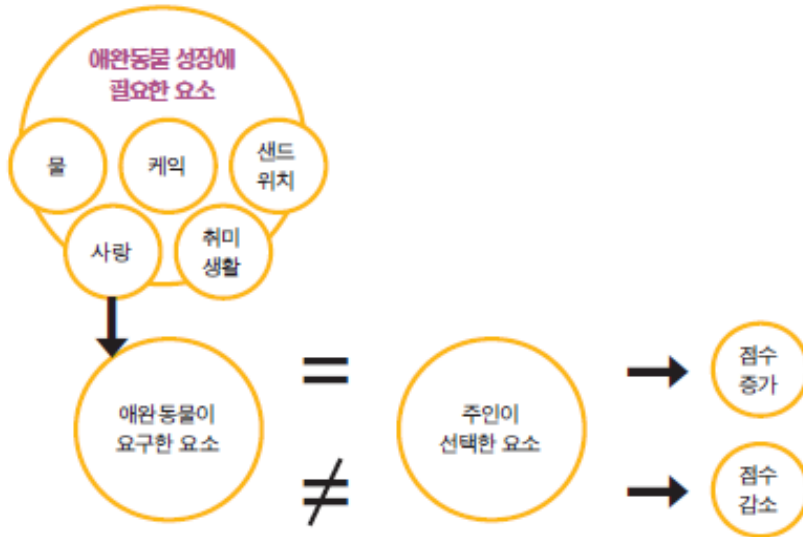
Discovery (탐구)

- 기존 애완동물 키우기 (소프트웨어 중등교재 06. 디지털 애완동물 키우기) 게임을 하면서 애완동물 성장에 사용된 요소를 찾아본다.
- 성장에 필요한 요소를 얻기 위한 알고리즘을 분석한다.

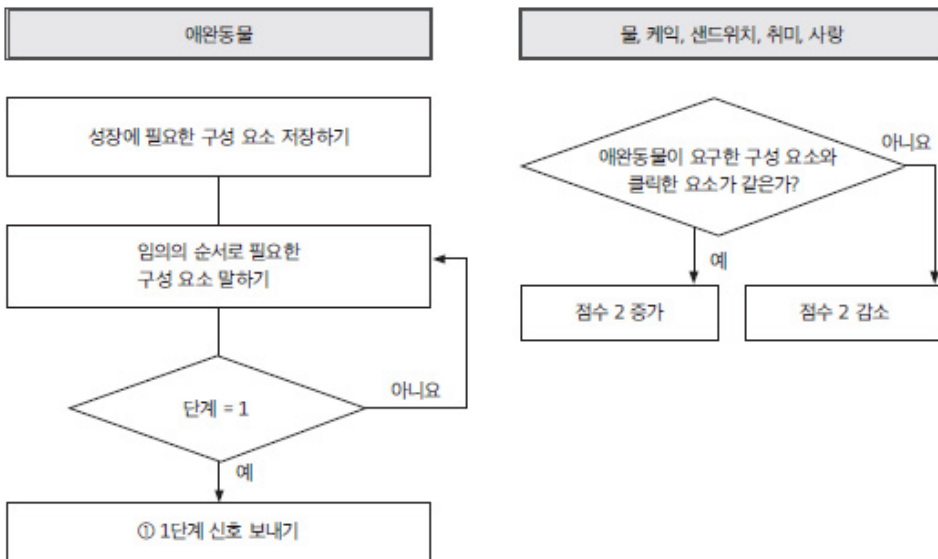


Design (설계)

- 성장에 필요한 요소를 얻는 방법을 설계해본다.

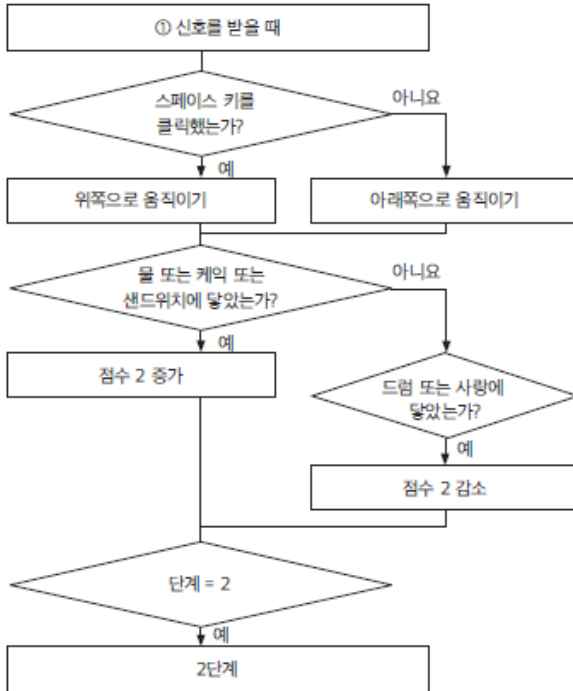


성장에 필요한 요소 제공하기



애완동물 움직이기

애완동물



Development (개발)

- 실제로 애완동물 키우기 프로그램을 제작해본다.



- 다른 그룹의 작품을 보고 비교해본 후 자신들의 프로그램을 수정하고 최종 작품을 완성한 후 공유하기로 수업을 마무리 한다.

4. 평가 계획

연번	평가 기준	방법
1	애완동물의 성장에 필요한 요소들을 파악하고 구체적인 프로그램 만들기 목표를 설정할 수 있는가?	관찰
2	디지털 애완동물 만들기 알고리즘을 설계하고 프로그래밍 언어를 통해 구현할 수 있는가?	동료평가
3	만들어진 프로그램을 발전시켜 새로운 프로그램을 만들 수 있는가?	수행평가

그림 그리기 앱 제작

적용학년	중학교
과목 및 차시	정보 2차시
수업모델	Discovery - Design - Development 모델
수업전략	협동학습, 디자인씹킹, 디버깅
도구 및 자료	앱인벤터, 구글 아이디, 포스트잇
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

그림 그리기 앱 제작에 D-D-D 모델을 적용하기 위해서는 그림을 그리는 데 필요한 기능들을 구현하기 위한 자료를 수집 및 조사하고, 개발하고자 하는 모듈과 일정한 패턴을 학생 스스로 탐구하도록 하여 추상화와 알고리즘의 핵심 내용을 파악하도록 합니다. 탐구결과 알게 된 것을 바탕으로 필요 컴포넌트, 컴포넌트의 특성 및 역할, 컴포넌트 간 상호작용 등을 이해하기 쉽게 순서도나 의사(pseudo)코드 등을 활용하여 설계합니다. 학생이 설계한대로 제작해 보고 서로 공유하여 피드백을 받고 디버깅하여 최종 작품을 완성합니다. 이런 일련의 과정을 통하여 CT 요소들을 통합적으로 경험해 볼 수 있습니다.

2. 학습 목표

- 앱인벤터를 활용하여 그림그리기 앱을 제작할 수 있다.

3. 수업 단계 및 활동

Discovery (탐구)	
<ul style="list-style-type: none"> - 그림그리기 앱에 필요한 기능들을 찾아보게 한다. (포스트잇 등을 사용하여 모듈원들의 생각을 모은다.) - 사용자가 편리하게 사용하기 위해 고려할 점에 대해 토론시킨다. - 색상 표현 방법에 대해 조사시키고 모듈별로 발표시킨다. - 앱인벤토리로 제작 시 사용할 컴포넌트들을 찾아보게 한다. 	

분해	<ul style="list-style-type: none"> - 그림그리기 앱에 필요한 기능들을 나열해본다. (색상선택, 붓크기조절, 부분지우기, 전체지우기, 투명도 조절, 캔버스 설정 등) - RGB 색상모드의 색 구성 요소별로 분해한다. - 앱의 동작을 단계별로 분해한다. (단계별 사용 컴포넌트를 알아본다.)
패턴인식	<ul style="list-style-type: none"> - 단계별 동작 패턴을 탐구한다. (예: 버튼 - 슬라이더 컴포넌트 조절 - 레이블 컴포넌트에 표시)
추상화	<ul style="list-style-type: none"> - 그림 그리기 도구에 대한 핵심 기능을 수학적 방법으로 공식화하기(예를 들어 직선그리기 도구의 경우 (x_1, y_1)좌표와 (x_2, y_2)좌표를 찾아 선으로 연결하는 공식을 활용한다)

Design (설계)	
<ul style="list-style-type: none"> - 그림그리기 앱에 필요한 컴포넌트를 배치해보고 구성 요소별 연관성을 고려하여 구상도를 설계한다. 	

알고리즘 활동	<ul style="list-style-type: none"> - 뷰어 창에 필용한 컴포넌트들을 배치한다. - 컴포넌트별로 해당 속성을 설정한다. - 컴포넌트들 간의 연계성을 순서도나 의사(pseudo)코드로 표현해본다.
---------	--

Development (개발)

- 학습내용을 토대로 하여 실제 프로토타입을 제작해보고 발표하여 피드백을 받는다.
- 다른 그룹의 작품을 보고 비교해본 후 자신의 프로그램을 디버깅하며 최종 작품을 완성한 후 공유한다.

프로그래밍 활동

- 디자인 단계의 작업에 바탕을 두고 블록 편집 모드에서 프로그래밍 하여 프로토타입을 제작한다.
- 모듈별로 결과물을 발표하고 피드백 받는다.
- 디버깅 과정을 거치며 설계와 구현을 반복한다.
- 최종 작품을 완성한 후 공유한다.

4. 평가 계획

연번	평가 기준	방법
1	그림 그리기 앱에 필요한 컴포넌트들을 구성할 수 있다.	퀴즈/관찰
2	RGB 색상모드의 색표현 방식을 설명할 수 있다.	관찰
3	바꾸고 싶은 요소를 찾아 변경해야 할 코드를 발견할 수 있는가?	수행평가
4	테스트와 디버깅을 거쳐 최종작품을 만들 수 있는가?	수행평가

실험실 지킴이

적용학년	고등학교
과목 및 차시	정보, 2차시
수업모델	Discovery - Design - Development 모델
수업전략	탐구학습, 협동학습
도구 및 자료	텍스트 기반 프로그래밍언어
중점 CT요소	<ul style="list-style-type: none"> ■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

D-D-D 모델로 학생 스스로 탐구하는 과정을 가질 수 있도록 교수학습 내용을 구성하였습니다. 실험실 지킴이에 D-D-D 모델을 적용하기 위해서 실습실 환경의 조건을 분석하고 입력, 처리, 출력으로 구분하여 과정별로 문제를 분해합니다. 학생 스스로 일정한 규칙의 패턴을 탐구하여 추상화와 알고리즘의 핵심 내용을 파악하고, 순서도나 의사(pseudo)코드 등으로 알고리즘을 표현한 뒤 프로그램을 구현, 자동화하고 공유와 수정을 거쳐 작품을 완성합니다. 이런 일련의 과정을 통해 컴퓨팅 사고를 통합적으로 경험할 수 있습니다.

2. 학습 목표

- 실험실의 제한된 조건을 해결하는 실험실 지킴이 프로그래밍을 제작할 수 있다.

3. 수업 단계 및 활동

Discovery (탐구)

<문제 제시>

음식물로 퇴비를 만드는 재활용 실험을 위해 과학시간에 음식물의 발효 조건을 온도 40℃, 습도 60%, 발효기간 30일로 제한하여 실험을 한다. 온도와 습도, 발효기간의 조건을 하나라도 유지하지 않으면 경고음을 내어 알려준다. 발효기간 30일 동안 실험실의 조건을 유지하도록 체크하는 시스템을 자동화 해보자.

(단, 경고음은 어떤 음이든 고려하지 않고, 실험 시작일이 발효기간의 시작일이 된다.)

- 문제를 분석하고 실험실 조건의 자동화를 위한 방법과 컴퓨팅의 원리 및 기능을 탐색해 본다.

- 문제 상황을 이해하고 주어진 조건과 해결할 목표를 분석한다.
- 실험실 조건의 입력 기능과 경고음을 발생시키는 기능을 조사한다.
- 사용할 프로그래밍 언어에서 문제 해결에 사용할 기능을 적용가능한지 조사한다.
- 탐색한 내용을 모둠별로 발표한다.

[분석활동] 해결할 목표를 찾고 조건을 분석해 보자.

해결 목표	발효기간 동안 실험실의 제한 조건의 온도, 습도를 유지한다.
조건 분석	<ul style="list-style-type: none"> - 온도40℃, 습도60%를 유지해야 한다. - 발효기간 30일을 경과하지 않아야 한다. - 경고음은 어떤 음이든지 무관하지만, 제한된 온도, 습도, 발효기간의 조건을 벗어나면 발생한다. - 실험 시작 일에서부터 발효기간이 체크된다.

- 문제를 분해하여 처리할 주요 요소를 추출하고 패턴을 찾아 추상화한다.

분해	<ul style="list-style-type: none"> - 실험실의 조건 처리 과정을 입력, 처리, 출력으로 나누어 구분하고 표, 그림, 글 등으로 표현한다. - 실험실의 온도, 습도, 발효기간의 조건을 고려하여 처리 가능한 단위의 작은 문제들로 나누어 본다. - 분할된 문제에서 필요한 주요 요소를 추출한다. - 아이디어와 의견을 모아 문제를 함께 분해해 본다.
패턴인식	<ul style="list-style-type: none"> - 실험실 조건의 처리 과정별 일정한 규칙의 처리 패턴을 탐구하여 나열한다. - 학생 스스로 탐구하여 추상화와 알고리즘의 핵심내용을 파악하도록 한다.
추상화	<ul style="list-style-type: none"> - 해결하기 쉬운 형태로 구조화한다. - 온도, 습도, 발효기간의 변수들 간의 관계를 명확하게 한다.

[분해활동] 입력, 처리, 출력 과정으로 구분하여 해결 문제를 찾아보자.

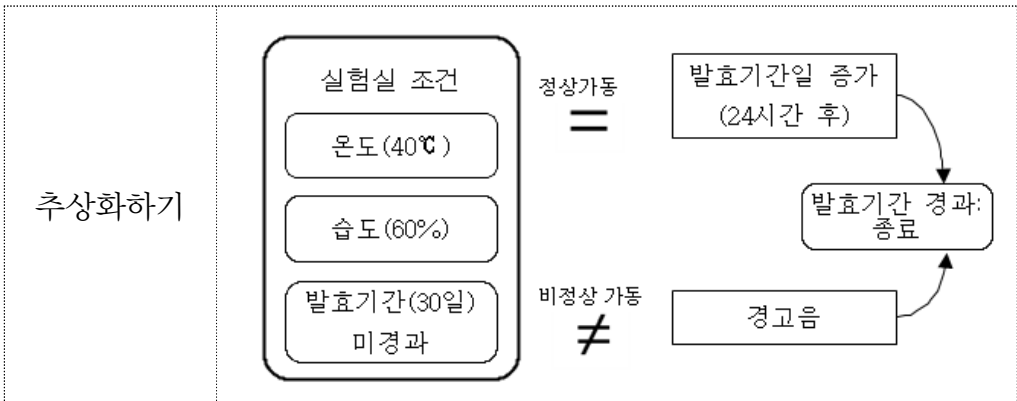
입력	<ul style="list-style-type: none"> - 온도: 40℃, 습도: 60%, 발효기간: 30일 - 실험 시작일(발효시작일)
처리	<ul style="list-style-type: none"> - 온도, 습도, 발효기간의 조건이 충족되거나 충족되지 않을 경우의 처리 - 발효기간이 경과하지 않을 때와 종료된 경우의 처리 - 여러 조건에 대한 경고음의 발생 처리
출력	<ul style="list-style-type: none"> - 온도나 습도가 맞지 않거나 발효기간이 경과되면 경고음 발생

[패턴인식활동] 실험실의 제한 조건의 처리를 위한 일정한 규칙의 처리 패턴을 찾아보자.

패턴 찾기	<ul style="list-style-type: none"> - 온도와 습도가 높을수록 발효 속도가 빨라진다. - 온도와 습도가 일정 조건의 이하 또는 이상인 경우 발효는 되지 않는다.
-------	--

[추상화활동] 실험실의 제한 조건에서 처리할 핵심요소를 추출하여 표, 그림, 그래프 등의 형태로 추상화해 보자.

핵심요소 추출	실험실 조건(온도(45℃) 습도(60%)), 발효기간(30일, 실험 시작일 기준)
---------	---



Design (설계)

- 알고리즘의 설계 방법을 조사하고 작성할 설계 방법을 선택한다.
- 구현할 알고리즘의 구체적인 설계는 순서도, 의사(pseudo)코드 등으로 알고리즘을 표현한다.

[활동] 문제해결 방법을 다양하게 만들고 실제 적용할 효과적인 방법을 찾아보자.

창의적 아이디어	〈아이디어1〉 〈아이디어2〉
해결 방법	〈방법1〉 〈방법2〉
	〈선택 방법〉 〈선택한 이유〉

[협력활동] 프로그램을 나누어 작성할 수 있도록 처리할 주요 기능을 나누어 보자.

기능 나누기	
--------	--

Development (개발)
<ul style="list-style-type: none"> - 실제 실험실 지킴이 프로그램을 구현한다. - 작품을 발표하고 다른 그룹과 비교 후 자신들의 프로그램을 수정하여 최종 작품을 완성한 후 공유한다.

4. 평가 계획

연번	평가 기준	방법
1	실험실 조건 만들기에 필요한 요소들을 파악하고 구체적인 프로그램 만들기 목표를 설정할 수 있는가?	발문/관찰
2	실험실 조건 만들기의 추상화와 효율적인 알고리즘을 설계할 수 있는가?	수행평가
3	프로그래밍 언어를 통해 구현할 수 있는가?	수행평가
4	공유 및 피드백을 통해 프로그램의 개선할 부분을 수정하여 프로그램을 발전시켜는 태도를 보이는가?	관찰 동료평가

물을 주는 화분 만들기

적용학년	6학년
과목 및 차시	실과
수업모델	Need-Design-Implement-Share 모델
수업전략	스토리텔링, 협동학습
도구 및 자료	블록 기반 프로그래밍언어, 피지컬컴퓨팅 도구, 포스트잇
중점 CT요소	■ 분해 □ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

교실에서 키우는 화분이 관리가 되지 않는 경우는 흔하게 발견할 수 있습니다. 왜 이러한 문제가 일어나는지 생각해보고 문제점을 개선하기 위하여 어떻게 하면 좋을지 생각해보는 활동으로 ‘물을 주는 화분 만들기’라는 주제를 선정하였습니다. 화분 속 수분 양과 같은 주변 환경의 변화에 따라 적절하게 물을 공급하거나 일정 시간이 지나면 물을 주는 화분이 있다면 화분이 더욱 싱싱하게 자랄 수 있을 것입니다. 학생들은 자신의 주변에서 일어나는 문제에 관심을 가지고 참여함으로써 자신의 창의적인 설계 능력을 발휘하여 알고리즘을 설계하고 구현함으로써 문제를 해결하고, 이를 서로 공유하고 일반화하는 과정을 통해 자기성찰을 할 수 있도록 구성하였습니다.

2. 학습 목표

- 주어진 문제를 이해하고 생활에 도움이 되는 해결 방법을 모색할 수 있다.
- 물을 주는 화분의 구조와 알고리즘을 설계할 수 있다.
- 프로그래밍 언어를 통해 구현해보고 이를 공유할 수 있다.

3. 수업 단계 및 활동

Needs (요구분석)

- 문제 상황

1 토마토 모종을 심었으니 다 같이 잘 관리해서 토마토가 자라는 모습을 잘 관찰하도록 해요.

네~!

2 처음에는 물도 자주 주고 많은 신경을 썼는데.. 관리가 잘 안 되는 것 같네..

3 우리가 요즘 물을 잘 안 주었더니 잎도 마르고 시들어 버린 것 같아.

물이 부족하면 자동으로 물을 줘서 식물을 관리해 주는 화분이 있던데.....

4 우리 반에도 물이 부족하면 자동으로 물을 주는 화분이 있었으면 좋겠다.

우리가 한번 만들어 볼까?

√ 이야기 형태로 주어진 문제 상황을 보고 문제의 발생 원인과 해결 방법을 모색한다.

√ 문제 상황은 토마토를 화분에 심었으나 관리가 잘 되지 않아 시들어 버린 상황이 발생한 것이다.

√ 우리가 편리하게 화분을 관리하려면 어떻게 해야 할지 고민해보고 경험이나 주어진 자료, 탐색 활동을 통해 자동으로 물을 주는 화분이 있다면 문제를 해결할 수 있을 것이라고 생각한다.

- 사용자 요구분석

- √ 문제 상황을 이해하고 자동으로 물을 주는 화분을 만들 수 있는지 생각한다.
- √ 자동으로 물을 주는 화분이 있다면 얼마나 편리해질지 생각한다.
- √ 물을 주는 화분을 만들기 위해 알아야 할 것이 무엇인지 이해하고 분석한다.
- √ 탐색한 내용을 모듬별로 발표한다.

[활동] 화분 속 물의 양에 따른 식물의 상태 변화를 알아보고, 알게 된 내용을 발표해 봅시다.

- 문제 이해 및 분석

- √ 문제를 이해하고 자동화의 주요 요인을 분석한다.
- √ 물의 양에 따라 물을 자동으로 줄 수 있는 원리와 방법을 탐색한다.
- √ 자동화에 필요한 자료를 조사한다.
- √ 목표를 찾고 조건을 분석한다.
- √ 탐색한 내용을 모듬별로 발표한다.




[활동] 문제를 분석하여 정리해 보자.

문제 상황의 이해	문제 해결 방법 만들기	필요한 자료의 수집
해결할 문제의 최종목표는 무엇인지 생각해 보자.	어떻게 문제를 해결할 수 있을지 아이디어를 모으고 해결 방법을 창의적으로 만들어 보자.	문제 해결에 필요한 자료는 어떤 것이 있을지 찾아보자.
	아이디어:	
	방법:	

Design (디자인)

<p>분해</p>	<ul style="list-style-type: none"> - 물을 주는 화분을 어떤 모양과 기능을 넣어 만들면 좋을지 생각한다. - 물을 양을 측정하는 기능, 물을 주는 기능 등으로 문제를 분해하고, 물을 주는 기능에서 물이 어느 정도일 때 물을 줄지 생각한다.
<p>패턴인식</p>	<ul style="list-style-type: none"> - 물의 양에 따라 물을 주는 화분의 동작 패턴을 찾아 나열한다. - 핵심 요소 간 패턴을 쉽게 파악할 수 있도록 시각화한다. - 시각화를 할 때는 창의성을 신장하기 위해 마인드 맵, 브레인스토밍, 그래프와 도식화 등의 다양한 전략을 사용하고, 생각을 손쉽게 추가, 수정, 삭제가 쉬운 포스트잇과 같은 도구를 사용한다.
<p>추상화</p>	<ul style="list-style-type: none"> - 식물이 잘 자라기 위한 온도와 습도의 조건을 공식으로 표현한다.
<p>알고리즘 설계</p>	<ul style="list-style-type: none"> - 물을 주어야 할 때와 주지 말아야 할 때 등 알고리즘의 핵심 요소를 추출한다. - 물의 주는 화분의 프로그램의 알고리즘을 생각해보고 의사(pseudo)코드나 순서도로 표현한다.

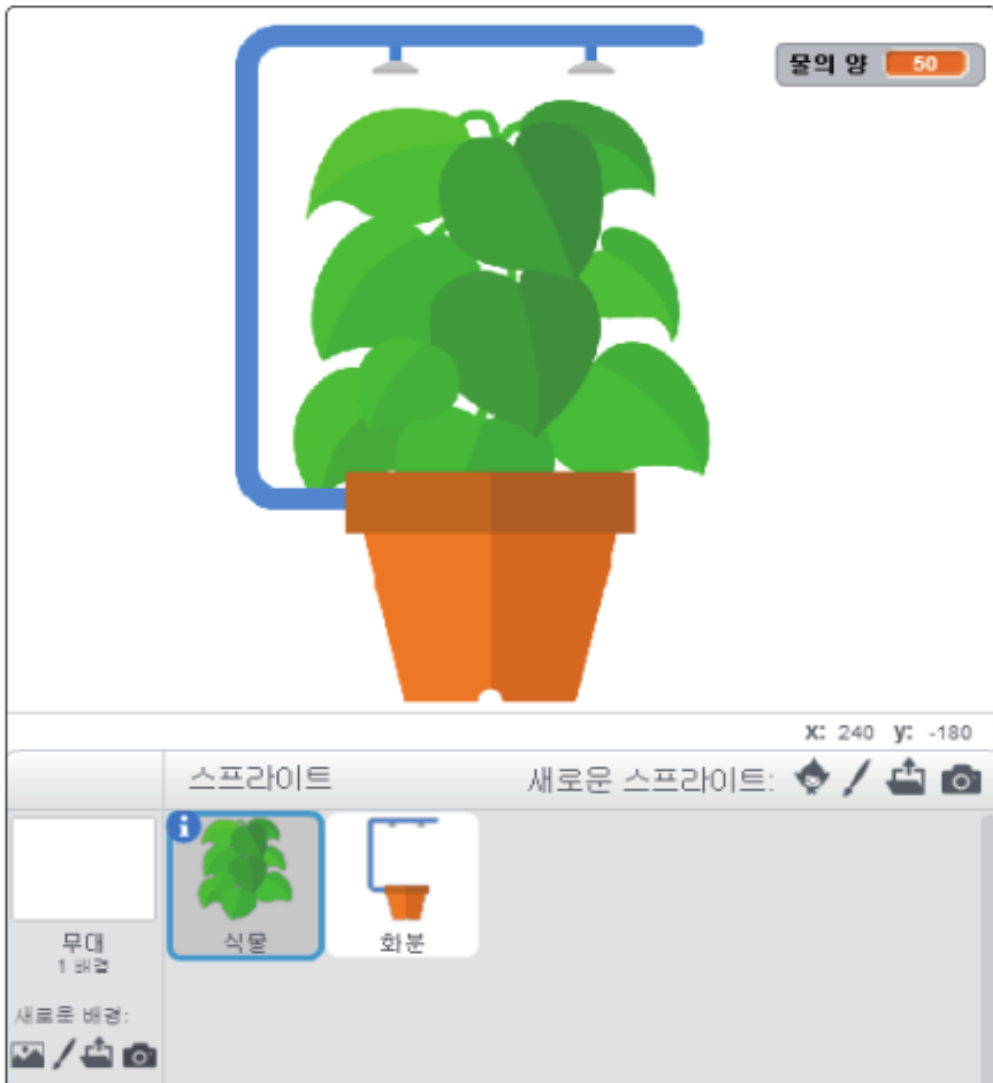
[활동] 물을 주는 화분의 알고리즘을 설계하여 봅시다.

대상	알고리즘
 <p>식물</p>	<p>계속 반복하기</p> <p>만일 () 값 < () 라면 (싱싱하다 / 시든다)</p> <p>아니면</p> <p>만일 () 값 > () 라면 (싱싱하다 / 시든다)</p> <p>아니면 (싱싱하다 / 시든다)</p> <p style="text-align: center;">보기</p> <p>• 물의 양 • 30 • 80 • 싱싱하다 • 시든다</p>
 <p>물을 주는 화분</p>	<p>계속 반복하기</p> <p>만일 () 값 < () 라면 만일 () 값 < () 인 동안 반복 (물 주기 / 물 주지 않기) () 값에 () 더하기</p> <p>아니면 (물 주기 / 물 주지 않기)</p> <p style="text-align: center;">보기</p> <p>• 물의 양 • 30 • 80 • 5 • 10 • 물 주기 • 물 주지 않기</p>
 <p>화분 속 물의 양</p>	<p>계속 반복하기</p> <p>() 값에서 () 만큼 빼기</p> <p style="text-align: center;">보기</p> <p>• 물의 양 • 5 • -5</p>

Implement (구현)

- EPL을 활용하여 가상의 화분을 만들고 프로그래밍하여 구현한다.
- 학습자의 수준과 환경에 따라 간단한 센서나 기구를 활용하여 실제 화분을 제작할 수 있다.

[활동] 물을 주는 화분 프로그램을 만들어 봅시다.



Share (공유)

- '물을 주는 화분' 프로그램을 발표하고 공유한다.
- 서로 좋은 점과 개선할 점에 대한 이야기를 나누고 피드백을 받는다.
- 개발 과정에 대한 간략한 소개 글을 작성하여 온라인에 탑재하고 댓글 등의 반응을 통해 자기성찰을 한다.

4. 평가 계획

연번	평가 기준	방법
1	주어진 문제를 이해하고 '물을 주는 화분'의 핵심 기능을 추출할 수 있는가?	관찰
2	물을 주는 화분의 알고리즘을 설계할 수 있는가?	수행평가
3	EPL을 통해 설계한 프로그램을 구현하고 공유할 수 있는가?	수행평가
4	공유 및 피드백을 하려는 태도를 보이는가?	관찰

에너지를 절약하는 똑똑한 선풍기

적용학년	3학년~6학년
과목 및 차시	창의적 체험활동, 실과 (2차시)
수업모델	Need-Design-Implement-Share 모델
수업전략	협동학습, 디버깅, 디지로그
도구 및 자료	EPL, 포스트잇, 피지컬컴퓨팅 도구
중점 CT요소	■분해 ■패턴인식 ■추상화 ■알고리즘 ■자동화

1. 개요

본 수업은 ‘에너지를 절약하는 똑똑한 선풍기’ 라는 주제를 디자인 씽킹 모델을 이용해서 진행하고자 하였습니다. 수업 초반에는 아무도 없는 교실에서 작동하고 있는 선풍기로 인한 전력난에 대한 문제에 대한 고찰과 선풍기 끄는 당번이 누구인지 모르는 사용자에게 대한 탐색을 합니다. 그 후 사람이 없을 때 자동으로 선풍기를 꺼주는 프로그램이 있으면 좋다고 생각하고, 사람의 움직임을 감지할 수 있는 방법을 알아보고 자동으로 선풍기를 꺼지게 하는 프로그램을 만드는 수업입니다.

이를 위하여 프로그래밍 언어로 구현하기 전에 포스트잇과 교재를 사용해 프로젝트의 스토리, 필요 객체, 객체의 특성 및 역할, 객체 간 상호작용 등을 이해하기 쉽게 계획합니다. 그 후 계획한 프로젝트를 EPL을 통해 구현합니다. 만일 피지컬 교구를 다룰 수 있는 고학년의 경우 아날로그 정보를 디지털 정보로 표현해 보는 피지컬 컴퓨팅 활동을 통해 융합 활동과 연계할 수 있습니다.

2. 학습 목표

- 사람이 없을 때 자동으로 선풍기를 꺼주는 프로그램을 만들 수 있다.

3. 수업 단계 및 활동

Needs (요구분석)
<ul style="list-style-type: none"> - 문제 파악하기 - 문제 정의하기 - 문제 해결방법을 생각해보기 - 문제 해결방법의 장단점 파악하기 - 생활 속에서 문제와 관련된 예제 찾기

위 대화를 읽고 해결해야 할 문제를 정리하여 아래 빈칸에 채워 봅시다.

누가 학급 친구들	언제 체육 시간이 끝나고 쉬는 시간에	어디에서 교실에서	왜 에너지가 낭비되므로
어떻게 교실에서 사람의  할 수 있는 방법을 알아보고	무엇을  없을 때 자동으로 선풍기를 꺼지게 하는 프로그램을 만들자		

〈Needs 단계〉

Design (디자인)
<ul style="list-style-type: none"> - 문제를 해결하기 위해 만들어진 내용 살펴보기 - 문제를 해결하기 위한 알고리즘 만들기 - 알고리즘 순서를 완성하기 - 알고리즘 완성하기 - 오브젝트 내용(움직임, 대사) 만들기



알고리즘을 만들어 보아요.

알고리즘을 만들어 봅시다.

대상	알고리즘
 사람	<p>() 번 반복</p> <ol style="list-style-type: none"> 1 2 교실 밖에서 4초 동안 기다리기 3 교실 안으로 2초 동안 움직이기 4 <p style="text-align: center;">보기</p> <ul style="list-style-type: none"> • 4번 • 교실 밖으로 2초 동안 움직이기 • 교실 안에서 4초 동안 기다리기
 감지 센서 범위	<p>() 반복</p> <ol style="list-style-type: none"> 1 '움직임 있음' 신호 보내기 만일 사람에게 닿지 않았다면 2 <p style="text-align: center;">보기</p> <ul style="list-style-type: none"> • 계속 • 만일 사람에게 닿았다면 • '움직임 없음' 신호 보내기

<Design 단계>

Implement (구현)

- 프로그래밍 준비하기
- 에너지 절약 선풍기 만들기
- 프로그래밍 구현하기
- 프로그램이 바르게 실행되었는지 O로 표시하기



〈Implementation 단계〉 <http://goo.gl/XcXeWf>

Share (공유)

- 구현한 프로그래밍 공유하기
- 구현한 프로그래밍 공유하기
- 공유 후 피드백 내용 성찰하기
- 제일 앞 단계로 돌아가기

4. 평가 계획

연번	평가 기준	방법
1	문제를 파악해서 정의하는가?	관찰
2	문제를 해결하기 위한 알고리즘을 세부적으로 제시하는가?	관찰/수행평가
3	문제를 해결하기 위한 프로그래밍을 바르게 구현되었는가?	관찰/수행평가
4	구현한 프로그램 피드백 및 환류를 하는가?	동료평가 수행평가

청소로봇 만들기

적용학년	중학교
과목 및 차시	정보 2차시
수업모델	Need-Design-Implement-Share 모델
수업전략	페어프로그래밍, 협동학습
도구 및 자료	블록 기반 프로그래밍언어, 로봇 등의 피지컬 도구
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

본 수업은 나만의 청소로봇을 만들어 보자는 주제로 청소 로봇이 있음으로 인간에게 어떤 도움을 줄 수 있는지를 생각해보는 과정입니다. 실제 시중에 판매되는 청소로봇의 알고리즘을 그림으로 그려보고 효과성을 분석하여 자료를 수집하도록 합니다. 이 후 다양한 모양의 공간에서도 청소가 가능하도록 창의적으로 설계해 보고 필요한 기능을 추출하여 나만의 청소 알고리즘을 설계하고, 실제로 로봇에 적용하여 시뮬레이션해보고 개발된 프로그램을 공유하며, 공유하여 얻은 피드백을 바탕으로 업그레이드 하는 과정을 통하여 컴퓨팅 사고력과 프로그래밍 능력을 증진시킬 수 있습니다.

2. 학습 목표

- 청소 로봇의 움직임에 필요한 규칙적인 패턴을 모델링 할 수 있다.
- 자신만의 청소 로봇의 동작 알고리즘을 고안하여 최적화 할 수 있다.

3. 수업 단계 및 활동

Needs (요구분석)

- 상황 제시를 통하여 청소로봇의 필요성을 이해한다.

1인 가족인 상범이는 집안 일 중에서 바닥 청소가 가장 싫다. 깨끗하게 청소를 해 놓으면 금방 먼지가 앉고 물걸레로 청소하기 위해서는 무릎이 너무 아프기 때문이다.

상범이는 인터넷을 통해 로봇 청소기 관련 내용을 보고 구매하려고 했으나, 가격이 너무 비쌌다. 그러던 중 'Fuzzbot'에 대한 기사를 접하고 이번 기회에 직접 로봇 청소기를 만들어 보기로 결심했다.

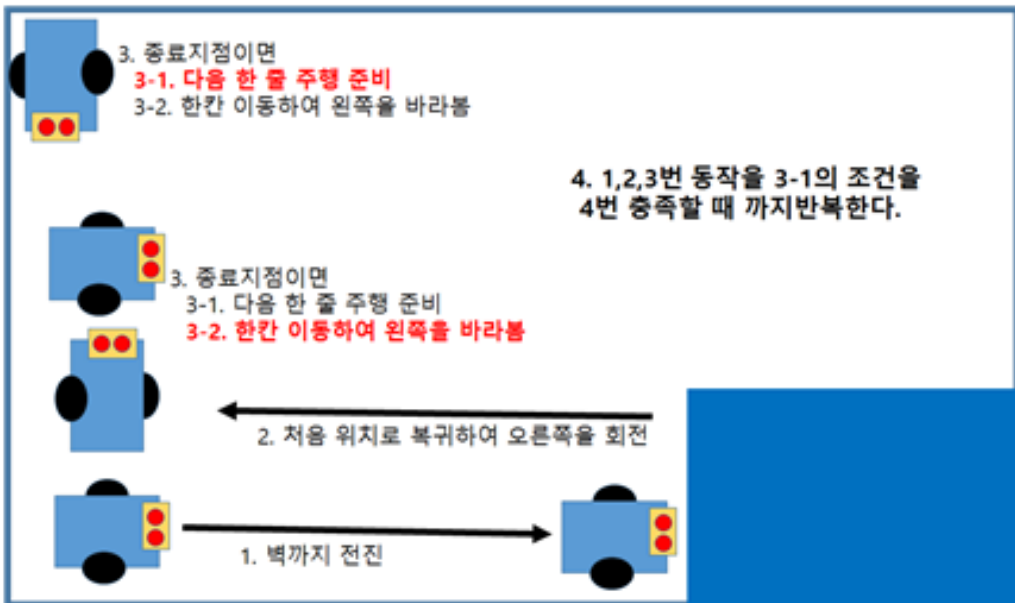
상범이는 쿤의 퍼즈봇의 꼬리 부분에 청소용 흡착포를 부착해서 움직이면 먼지가 제거되는 로봇을 만들고자 한다. 이제부터 나만의 청소 로봇을 만들어보자.

- 일반적인 로봇 청소기가 수행해야 할 기능이 무엇인지를 도출한다.
예) 먼지제거, 자동이동, 충전 등
- 로봇 청소기의 기능 중 만들고자 하는 로봇의 기능에 대하여 정의한다.
예) 어떤 방식으로 이동하여 청소할 것인가?



Design (디자인)

패턴인식	- 전체는 한 칸 이동을 전체만큼 반복하는 패턴을 찾아내도록 지도한다.
추상화	- 바둑판처럼 되어 있는 바닥을 청소하며 로봇이 움직일 때는 한 칸씩만 움직이도록 단순화한다.(후진 기능은 없음)
알고리즘	- 한 칸씩 이동하고 벽을 만나면 회전하여 처음 위치로 복귀하고 계속 이동한다.



Implement (구현)

- 설계된 알고리즘을 바탕으로 프로그램으로 구현하여 알고리즘대로 로봇이 이동하는지 확인한다.
- 프로그램을 실행해보고 오류사항을 발견하게 되면 디버깅을 수행하도록 지도한다.

- 실제 프로그램 예

[벽까지 전진하기]

```
                <소스코드>
void gotoWall( )
{
    while(getUltraSonic() > 10)
    {
        leftMotor(MOTOR_SPEED);
        rightMotor(MOTOR_SPEED);
    }
    move(0,0,DELAY_TIME);
}
```

[벽까지 전진했다가 처음 위치로 복귀하고 오른쪽으로 회전]

```
                <소스코드>
void shuttle()
{
    gotoWall( );
    //CCW 180 turn
    move(-MOTOR_SPEED,MOTOR_SPEED,TURN_TIME_180);
    move(0,0,DELAY_TIME);

    gotoWall( );
    //CW 90 turn
    move(MOTOR_SPEED,-MOTOR_SPEED,TURN_TIME_90);
    move(0,0,DELAY_TIME);
}
```

[앞 쪽이 벽(종료 조건)인지 판단하기]

<소스코드>

```
void loop() {
  int count=0;
  while(count<4)
  {
    shuttle();
    if(getUltraSonic() < 10) //end 1 corner
    {
      //CW 90 turn
      move(MOTOR_SPEED,MOTOR_SPEED,TURN_TIME_180);
      move(0,0,DELAY_TIME);
      //CCW 180 turn
      move(-MOTOR_SPEED,MOTOR_SPEED,TURN_TIME_180);
      move(0,0,DELAY_TIME);
      count++;
    }
    else //not yet end 1 corner
    {
      //CW 90 turn
      move(MOTOR_SPEED,-MOTOR_SPEED,TURN_TIME_90);
      move(0,0,DELAY_TIME);
    }
  }
}
```

Share (공유)

- 모듈별로 완성된 프로그램을 발표하고 공유해본다.
- 좋은 점과 개선할 점을 이야기하고, 여기서 나온 피드백을 중심으로 개선점을 반영하여 구현해 본다.
예) '로봇의 청소속도(이동속도)를 빠르게 해본다' 등
- 개발 과정에 대한 간략한 소개 글을 작성하여 온라인에 탑재하고 댓글 등의 반응을 통해 자기성찰을 한다.
- 개발과정에서의 자기 성찰을 해보고 몇몇 학생들에게 발표하도록 한다.

4. 평가 계획

연번	평가 기준	방법
1	청소로봇의 필요성을 인식하고 시중의 청소로봇의 알고리즘을 그림으로 표현할 수 있는가?	관찰
2	청소 로봇 주행하기 동작 원리를 이해하고 효과적인 주행 알고리즘을 설계할 수 있는가?	수행평가
3	EPL을 통해 설계한 프로그램을 구현하고 공유할 수 있는가?	수행평가
4	공유 및 피드백을 하려는 태도를 보이는가?	관찰

스마트한 가정환경 만들기

적용학년	고등학교
과목 및 차시	정보 2차시
수업모델	Need-Design-Implement-Share 모델
수업전략	협동학습, 액션러닝
도구 및 자료	텍스트 기반 프로그래밍언어, 피지컬 컴퓨팅 도구(빛 센서, LED 등)
중점 CT요소	■ 분해 ■ 패턴인식 ■ 추상화 ■ 알고리즘 ■ 자동화

1. 개요

이 주제는 햇빛의 밝기에 따라 커튼의 닫힘과 열림, 조명의 켜짐과 꺼짐에 대한 관계를 이해하여 인간을 중심으로 분석, 사고의 확장, 창의적인 아이디어를 산출하는 다양한 활동을 할 수 있습니다.

커튼과 조명의 자동화에 필요한 요소를 분해하여 패턴을 찾아 시각화하고 알고리즘을 설계하여 프로그래밍하는 전체 과정을 이해할 수 있고, 햇빛의 밝기를 빛 센서를 이용하여 입력받도록 설계하여 피지컬 컴퓨팅을 경험할 수 있습니다.

더 나아가 다른 그룹과 프로그램을 공유하고 피드백을 반영하여 최종 작품으로 완성해 나가며 개발과정에 대한 자기성찰을 할 수 있는 과정을 통해 컴퓨팅 기반의 문제 해결력을 더 향상시킬 수 있습니다.

2. 학습 목표

- 빛의 밝기의 변화에 따라 커튼과 조명을 동작시키는 프로그래밍을 할 수 있다.

3. 수업 단계 및 활동

Needs (요구분석)

- 문제 상황

장기간 출장 간 사람들이나 직장인들이 집에 늦게 돌아올 때 안정감을 주기 위해 햇빛의 양에 따라 커튼을 치고 조명이 켜지게 자동으로 동작하는 커튼과 조명을 만들려고 한다.

집안의 빛의 밝기가 일정량 이하로 줄어들면 커튼이 닫히고 증가하면 열린다. 조명은 에너지를 절약하기 위해 커튼이 닫힐 때 5시간 동안만 켜지고 꺼져야 한다. 생활의 편리함을 추구하도록 센싱 기술과 융합하여 해결해 보자(단, 커튼은 화면상으로 구현할 수 있다).

- 사용자 요구분석

- ✓ 문제 상황을 이해하고 자동화가 가능한 것인지 살핀다.
- ✓ 자동화가 인간에게 얼마나 이로움을 가져다주는지 그 가치에 탐색한다.
- ✓ 자동화를 통해 구해야 할 것이 무엇인지 이해하고 분석한다.
- ✓ 탐색한 내용을 모듈별로 발표한다.

[활동] 문제를 해결할 때 인간의 편리성과 이로움뿐만 아니라 발생하는 문제점에 대해서도 자신의 생각을 포스트잇에 적어 공유해 보자.

[활동] 조명이 켜지고 커지는 시간을 5시간으로 제한하는 이유를 탐색해보자.

- 문제 이해 및 분석

- √ 문제를 이해하고 자동화의 주요 요인을 분석한다.
- √ 커튼과 조명을 자동으로 조절하는 원리와 방법을 탐색한다.
- √ 자동화에 필요한 자료를 조사한다.
- √ 목표를 찾고 조건을 분석한다.
- √ 탐색한 내용을 모듬별로 발표한다.

[활동] 문제를 분석하여 정리해 보자.

문제 상황의 이해	문제 해결 방법 만들기	필요한 자료의 수집
해결할 문제의 최종목표는 무엇인지 생각해 보자.	어떻게 문제를 해결할 수 있을지 아이디어를 모으고 해결 방법을 창의적으로 만들어 보자.	문제 해결에 필요한 자료는 어떤 것이 있을지 찾아보자.
	아이디어: 방법:	

Design (디자인)

분해	<ul style="list-style-type: none"> - 컴퓨팅 사고를 기반으로 작은 단위의 문제로 분해한다. - 빛과 커튼, 조명과의 상호 동작 관계를 파악한다. - 빛의 밝기에 따라 처리 가능한 작은 단위의 문제로 분해하여 필요한 요소를 추출하여 글이나 표, 그림 등으로 구조화 해 본다.
패턴인식	<ul style="list-style-type: none"> - 빛의 밝기에 따른 커튼과 조명 요소 간의 패턴을 쉽게 파악할 수 있도록 시각화한다. - 시각화를 할 때는 창의성을 신장하기 위해 마인드 맵, 브레인스토밍, 그래프와 도식화 등의 다양한 전략을 사용하고, 생각을 손쉽게 추가, 수정, 삭제가 쉬운 포스트잇과 같은 도구를 사용한다.
추상화	<ul style="list-style-type: none"> - 빛의 밝기에 따라 변화되는 커튼과 조명의 동작 패턴을 공식으로 표현한다.
알고리즘 설계	<ul style="list-style-type: none"> - 알고리즘의 핵심 내용을 파악한다. - 구현할 알고리즘을 설계하여 순서도, 의사(pseudo)코드 등으로 표현한다.

[분해활동] 집안의 빛 양에 따라 커튼과 조명의 열고 닫힘, 켜짐과 꺼짐의 상관관계를 찾는 소그룹 활동을 해보자.

[분해활동] 문제를 분해하여 작은 단위의 문제로 만들어 보자.

[패턴인식] 패턴을 찾아 그림이나 표 등으로 시각화하고 발표해 보자.

[추상화활동] 발견한 패턴을 공식으로 유도하기 위한 방법을 논의하고 공식으로 표현하자.

[알고리즘 활동] 문제 해결을 위한 알고리즘을 표현해 보자.

Implement (구현)

- 센서나 기타 필요한 교구를 이용하여 실제 상황과 비슷한 환경을 만들어 제작한다.
- 실제로 커튼과 조명을 자동으로 조절하는 프로그램을 구현한다.

Share (공유)

- 프로그램을 발표하고 공유한다.
- 다른 그룹의 작품과 비교하여 자신들의 프로그램을 수정하고 교사의 피드백을 받아 최종 작품을 완성한다.
- 완성된 작품의 공유하고 자기성찰을 한다.

4. 평가 계획

연번	평가 기준	방법
1	빛의 밝기에 따라 커튼과 조명의 동작관계를 작은 단위의 문제로 분해할 수 있는가?	관찰
2	빛의 어두워지면 커튼이 닫히고 조명이 켜지는 알고리즘을 표현할 수 있는가?	수행평가
3	프로그래밍 언어로 구현할 수 있는가?	수행평가
4	공유 및 피드백을 통해 프로그램의 개선할 부분을 수정하여 반영하려는 태도를 보이는가?	관찰 동료평가

부록 2 SW교육 운영지침과 2015 개정 교육과정 비교

1) 초등학교 SW교육 운영지침과 2015 실과 교육과정 비교

초등학교 수준에서 소프트웨어교육 운영지침은 크게 생활과 소프트웨어, 알고리즘과 프로그래밍 2개 영역으로 제시되어 있으며, 이를 다시 나와 소프트웨어, 정보윤리, 문제해결 과정의 체험, 알고리즘의 체험, 프로그래밍의 체험 5개 중영역으로 세분화하여 11개의 내용 요소와 각각의 성취기준을 제시하고 있다. 이는 소프트웨어교육 운영지침이 교과 활동 외의 창의적 체험 활동을 통해 SW 교육에 내실을 기하기 위한 조치라고 볼 수 있다.

2015 개정 실과 교육과정의 성취기준과 비교하여 살펴보면

첫째, 운영지침의 ‘생활과 소프트웨어’ 영역의 ‘소프트웨어와 생활 변화’ 내용이 교과 활동으로 제시되고, 정보윤리와 관련된 사이버 예절, 개인정보 보호와 저작권 보호, 인터넷 중독 내용은 ‘기술 활용’ 영역에서 간략하게 제시되어 있다. 이는 정보윤리 관련 교육은 주로 창의적 체험활동을 통해 가르치도록 제시된 것으로 받아들일 수 있을 것이다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취 기준	내용 요소	성취 기준
소프트웨어와 생활변화	소프트웨어가 가져온 생활 모습의 변화를 설명할 수 있다.	소프트웨어의 이해	SW가 적용된 사례를 찾아보고 우리 생활에 미치는 영향을 이해한다.
사이버 공간에서의 예절	사이버 공간에서 지켜야 하는 예절을 알고 실천할 수 있다.		
인터넷 중독과 예방	게임과 인터넷 중독의 문제점을 알고 예방 방법을 설명할 수 있다.	개인 정보와 지식 재산 보호	사이버 중독 예방, 개인 정보 보호 및 지식 재산 보호의 의미를 알고 생활 속에서 실천한다.
개인 정보보호	개인 정보가 중요한 이유와 지킬 수 있는 방법을 찾아 발표할 수 있다.		
저작권 보호	생활 속에서 저작권의 보호를 받는 것은 어떤 것인지 찾아보고, 보호하기 위한 방법을 설명할 수 있다.		

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취 기준	내용 요소	성취 기준
문제의 이해와 구조화	제시된 문제를 이해할 수 있다. 제시된 문제를 단순화할 수 있다.	절차적 문제해결	절차적 사고에 의한 문제 해결의 순서를 생각하고 적용한다.
문제 해결 방법 탐색	문제를 해결하기 위한 방법을 순서에 따라 설명할 수 있다. 제안한 문제 해결 방법의 문제점과 개선 방법에 대해 설명할 수 있다.		
알고리즘의 개념	알고리즘의 개념을 이해할 수 있다.		
알고리즘의 체험	순차, 선택, 반복구조를 이용하여 문제해결 절차를 그림이나 기호를 이용하여 표현할 수 있다. 간단한 알고리즘(정렬, 탐색)을 체험활동을 통하여 이해할 수 있다.		
프로그래밍의 이해	프로그래밍 언어의 기본요소를 알 수 있다. 주어진 프로그램을 동일하게 만들 수 있다.	프로그래밍 요소와 구조	프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험한다. 자료를 입력하고 필요한 처리를 수행한 후 결과를 출력하는 단순한 프로그램을 설계한다.
프로그래밍의 체험	주어진 프로그램을 수정하여 자신만의 프로그램을 만들 수 있다. 내가 생각한 간단한 프로그램을 만들 수 있다.		문제를 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.
		로봇의 기능과 구조	생활 속에서 로봇 활용 사례를 통해 작동 원리와 활용 분야를 이해한다. 여러 가지 센서를 장착한 로봇을 제작한다.

둘째, 운영지침의 ‘알고리즘과 프로그램’ 영역의 ‘문제해결 과정의 체험’을 절차적 문제 해결 내용 요소로 제시하였고, ‘알고리즘의 체험’ 관련 내용은 프로그래밍 요소와 구조 내용 요소의 ‘문제 해결 프로그래밍 과정에서 순차, 선택,

반복 등의 구조를 이해한다'는 성취 기준에 일부 반영되어 있다.

셋째, 운영지침의 '프로그래밍의 이해와 체험' 내용은 '프로그래밍 요소와 구조'라는 내용 요소로 수정되어 제시되었으며, 성취 기준에 '기초적인 수준의 프로그래밍 과정의 체험과 단순한 프로그램의 설계' 정도로 반영되어 있다.

넷째, 실과 '기술 활용' 영역에 로봇과 관련한 교육내용과 성취기준이 마련되어 있다. 이는 로봇을 활용한 SW교육, 즉 피지컬 컴퓨팅을 교육 내용에 포함시킨 것으로 생각할 수 있다. 한편 성취기준의 '로봇을 제작'한다는 문구가 로봇을 조립하는 것으로 혼동될 수 있으나, 성취기준 해설을 통해 '로봇에 사용되는 센서와 인체 구조를 비교하여 이해하고 센서를 장착한 로봇을 제작하여 구동시켜 봄으로써 로봇의 작동 원리를 강조한다'고 명시하여 단순한 교구 조립이 아님을 명시하고 있다.

2) 중학교 SW교육 운영지침과 2015 정보과 교육과정 비교

2015 개정 교육과정 중등 정보 교과 교육 목표는 ‘정보윤리의식, 정보보호능력, 정보기술활용능력을 기르고 컴퓨터과학의 기본 개념과 원리, 컴퓨팅 기술을 바탕으로 실생활 및 다양한 학문 분야의 문제를 창의적이고 효율적으로 해결하는 능력과 협력적 태도를 기르는 데 중점을 둔다.’고 제시되어 있다.

먼저 중학교 정보 교과에는 운영지침의 ‘기본’ 과정을 중심으로 학생의 발달 수준과 학습 난이도 등을 고려하여 학습 부담을 줄이고, 선택 과정으로 제시되었던 정보 과목이 필수로 전환되면서 수업 시수가 68차시에서 34차시로 줄어든 것을 고려한 것이라 볼 수 있다. 예를 들어 운영지침 성취 기준에 제시된 선형 구조와 비선형 구조 개념이라든가 자연어, 순서도 등의 알고리즘 표현법, 정렬과 탐색 등 다양한 알고리즘 등이 제외되어 학습 부담을 완화하고자 한 것으로 보인다.

다음으로 성취기준을 보다 구체화하여 수업을 준비하는 데 구체적인 지침으로 활용할 수 있도록 수정하여 제시한 것으로 보인다. 내용 영역에 있어서 운영지침의 생활과 소프트웨어, 알고리즘과 프로그래밍, 컴퓨팅과 문제해결 3개 영역으로 제시된 반면 정보 교과에서는 정보문화, 컴퓨팅 시스템, 자료와 정보, 추상화와 알고리즘, 프로그래밍 5개 영역으로 보다 구체적으로 제시되었다. 예를 들어 운영지침에 ‘정보의 유형을 구분하고 활용할 수 있다’, ‘주어진 문제를 이해하고 분석할 수 있다’고 제시된 성취 기준이 각각 ‘인터넷, 응용 소프트웨어 등을 활용하여 문제 해결을 위한 자료를 수집하고 관리한다’, ‘실생활 문제 상황에서 문제의 현재 상태, 목표 상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석한다’ 로 구체적으로 표현되어 있다.

마지막으로 컴퓨팅 시스템 영역에 센서를 이용한 자료 처리 및 동작 제어 프로그램 구현 내용이 추가된 것은 피지컬 컴퓨팅과 관련된 내용이 개정 교육과정에 반영된 것이라 할 수 있다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
소프트웨어 종류와 특징	다양한 분야에서 활용되는 소프트웨어의 종류와 특징을 설명할 수 있다.		
소프트웨어의 활용과 중요성	소프트웨어의 사용이 실생활을 어떻게 변화시켰는지 이해하고 소프트웨어의 중요성을 설명할 수 있다. 소프트웨어와 관련된 다양한 진로와 직업을 찾아보고 발표할 수 있다.	정보사회의 특성과 진로	정보기술의 발달과 소프트웨어가 개인의 삶과 사회에 미친 영향과 가치를 분석하고 그에 따른 직업의 특성을 이해하여 자신의 적성에 맞는 진로를 탐색한다.
개인정보 보호와 정보 보안	개인정보의 개념을 이해하고 유형을 설명할 수 있다. 개인정보 침해에 대한 피해 사례를 조사하여 예방 방법을 사례별로 설명할 수 있다. 악성 프로그램의 의미를 알고 보안 소프트웨어를 설치하여 컴퓨터를 보호할 수 있다.	개인정보와 저작권 보호 / 사이버 윤리	정보사회 구성원으로서 개인정보와 저작권 보호의 중요성을 인식하고 개인정보 보호, 저작권 보호 방법을 실천한다.
지적재산의 보호와 정보 공유	지적 재산의 개념을 이해하고 보호하는 방법을 설명할 수 있다. 지적 재산권을 침해하지 않고 올바른 방법으로 정보를 공유할 수 있다.		
컴퓨터의 구성	컴퓨터의 구성 요소와 기능을 이해하고 컴퓨터가 자료를 처리하는 과정을 설명할 수 있다. 다양한 운영체제의 종류와 특징을 알 수 있다.	컴퓨팅 기기의 구성과 동작 원리*	컴퓨팅 시스템을 구성하는 하드웨어와 소프트웨어의 역할을 이해하고 유기적인 상호 관계를 분석한다.
네트워크와 정보 교류	네트워크를 바탕으로 정보를 공유하고, 교류할 수 있는 다양한 서비스를 알 수 있다.	센서기반 프로그램 구현*	센서를 이용한 자료 처리 및 동작 제어 프로그램을 구현한다.
정보의 유형	자료와 정보의 개념을 이해하고 정보의 디지털 표현 방법을 설명할 수 있다. 정보의 유형을 구분하고 활용할 수 있다.	자료의 유형과 디지털 표현	디지털 정보의 속성과 특징을 이해하고 현실 세계에서 여러 가지 다른 형태로 표현되고 있는 자료와 정보를 디지털 형태로 표현한다. 인터넷, 응용 소프트웨어 등을 활용하여 문제 해결을 위한 자료를 수집하고 관리한다.
정보의 구조화	선형 구조의 개념을 이해하고 설명할 수 있다. 비선형 구조의 개념을 이해하고 설명할 수 있다. 효율적으로 정보를 제공하고 처리하기 위해 서 리스트, 계층, 테이블, 다이어그램 등과 같은 다양한 형식으로 정보를 구조화할 수 있다.		실생활의 정보를 표, 다이어그램 등 다양한 형태로 구조화하여 표현한다.

* 컴퓨팅 시스템 영역의 내용 요소, 피지컬 컴퓨팅이 핵심내용으로 포함되었음.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
문제 해결 절차의 이해	문제를 해결하는 절차를 설명할 수 있다.	문제이해	실생활 문제 상황에서 문제의 현재 상태, 목표 상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석한다.
문제 분석과 구조화	주어진 문제를 이해하고 분석할 수 있다. 실생활의 문제를 구조화하여 표현할 수 있다. 실생활의 문제를 추상화하여 표현할 수 있다.		문제 해결에 필요한 요소와 불필요한 요소를 분류한다.
문제 해결 전략의 탐색	주어진 문제를 해결하기 위한 다양한 해결 방법을 찾을 수 있다. 실생활의 사례와 연계하여 알고리즘이 무	핵심요소 추출	문제 해결을 위한 다양한 방법과 절차를 탐색하고 명확하게 표현한다.
알고리즘의 이해	엇인지 그 의미와 중요성을 알 수 있다. 알고리즘이 갖추어야 할 조건을 이해하고 다양한 알고리즘을 비교할 수 있다. 여러 가지의 알고리즘 표현 방법을 이해하고 설명할 수 있다.		논리적인 문제 해결 절차인 알고리즘의 의미와 중요성을 이해하고 실생활 문제의 해결과정을 알고리즘으로 구상한다.
알고리즘의 설계	문제 해결 절차를 여러 가지 알고리즘 표현법(자연어, 의사 코드, 순서도 등)으로 나타낼 수 있다. 실생활에서 발생하는 여러 문제들을 순차, 선택, 반복 구조를 활용하여 알고리즘을 설계할 수 있다. 정렬과 탐색 등 다양한 알고리즘을 바탕으로 실생활의 여러 가지 문제를 해결할 수 있다.	알고리즘의 이해와 표현	
프로그래밍 언어의 이해	프로그래밍 언어의 개념과 종류를 설명할 수 있다. 프로그래밍 언어의 개발 환경을 이해할 수 있다.		프로그래밍 언어의 개발 환경 및 특성을 이해한다.
프로그래밍의 기초	자료의 입출력문을 작성할 수 있다.	입력과 출력	다양한 형태의 자료를 입력 받아 처리하고 출력하기 위한 프로그램을 작성한다.
	문제 해결에 필요한 순차, 선택, 반복 구조를 프로그래밍할 수 있다.	제어구조	순차, 선택, 반복의 개념과 원리를 이해하고 세 가지 구조를 활용한 프로그램을 작성한다.
	여러 가지 구조적 절차를 복합적으로 구현하여 문제 해결에 활용할 수 있다. 비교-논리 연산자와 선택, 반복 구조를 활용하여 프로그래밍할 수 있다. 프로그램에서 오류를 확인하여 수정할 수 있다.	변수와 연산	변수의 개념을 이해하고 변수와 연산자를 활용한 프로그램을 작성한다.
실생활 문제 해결	실생활 문제를 알고리즘 설계와 프로그래밍으로 해결할 수 있다.	프로그래밍 응용	실생활 문제 해결을 위한 소프트웨어를 협력하여 설계, 개발, 비교-분석한다.
다양한 영역의 문제 해결	다양한 영역의 문제를 알고리즘 설계와 프로그래밍으로 해결할 수 있다.		

3) 고등 SW교육 운영지침과 2015 정보과 교육과정 비교

고등학교 정보과는 심화 선택에서 기술·가정 및 정보 가운데 한 가지 과목을 선택하도록 변경되었다. 심화 선택에서 일반 선택과목으로 변경되면서 향후 상당수의 학교에서 정보 과목을 선택할 것으로 기대된다. 고등학교 정보과 역시 정보문화, 컴퓨팅 시스템, 자료와 정보, 추상화와 알고리즘, 프로그래밍 5개 영역으로 제시되었다. 고등학교 정보 교과에서는 지식 중심의 내용 요소는 눈에 띄게 축소되고 문제해결과 프로그래밍 관련 내용 요소와 성취기준은 상당 부분 명확하게 제시하고자 한 것으로 보인다. 이는 교육과정을 개정한 본연의 취지를 달성하고자 하는 노력으로 여겨진다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
컴퓨팅 기술과 융합	컴퓨터 기술을 이해하고 다양한 컴퓨터 기술 분야를 찾을 수 있다. 컴퓨터 기술을 다른 학문과 융합한 예를 설명할 수 있다.	정보사회에서 정보과학의 지식과 기술이 활용되는 분야를 탐색하고 영향력을 평가한다.	
소프트웨어 의 미래	과거와 현재의 여러 분야에서 소프트웨어가 사용됨으로써 변화된 생활의 모습을 설명할 수 있다. 다양한 분야에서 소프트웨어가 사용되는 미래 발전 방향을 예측할 수 있다.	정보과학과 진로 정보과학 분야의 직업과 진로를 탐색한다.	
정보윤리와 지적 재산 보호	정보 윤리와 관련된 법과 제도를 살펴보고 정보 윤리 의식을 가질 수 있다. 지적 재산권을 침해하지 않고 저작물을 올바르게 이용할 수 있다.	저작권 활용 / 사이버 윤리	소프트웨어 저작권 보호 제도 및 방법을 알고 올바르게 활용한다.
정보보안과 대응 기술	정보보안에 대한 개념을 이해하고 대응 방법을 설명할 수 있다. 정보보안 기술을 활용하여 정보를 보호할 수 있다.	정보보호와 보안	정보보호 제도 및 방법에 따라 올바르게 정보를 공유하는 방법을 실천한다. 정보보안의 필요성을 이해하고 암호 설정, 접근 권한 관리 등 정보보안을 실천한다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
	컴퓨터의 구성 요소와 동작 원리를 설명할 수 있다.	운영체제 역할*	문제 해결에 적합한 하드웨어를 선택하여 컴퓨팅 장치를 구성한다.
정보기기의 동작 원리	운영체제의 관리 기법을 이해하고 설명할 수 있다.		운영체제의 개념과 기능을 이해하고 운영체제를 활용하여 컴퓨팅 시스템의 자원을 효율적으로 관리한다.
	네트워크의 전송원리를 이해하고 상황에 맞게 활용할 수 있다.	네트워크 환경 설정*	유무선 네트워크의 특성을 이해하고 사용하는 컴퓨팅 시스템의 네트워크 환경을 설정한다.
정보처리의 과정	정보의 다양한 처리 과정을 설명할 수 있다.	피지컬 컴퓨팅 구현*	피지컬 컴퓨팅 장치의 동작을 제어하기 위한 프로그램을 작성한다.

* 컴퓨팅 시스템 영역의 내용 요소

먼저 SW교육 운영지침의 ‘컴퓨팅과 정보 생활’ 중영역에서는 컴퓨팅 기술과 융합, 소프트웨어의 미래라는 내용 요소 각각에 2가지의 성취기준을 제시하고 있으나, 2015 개정 교육과정에는 ‘정보과학과 진로’라는 하나의 내용 요소와 2가지의 성취 기준으로 압축적으로 표현되어 있다. 한편 컴퓨팅 시스템 영역의 성취기준에 ‘피지컬 컴퓨팅’이 명시되었다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
정보의 표현	정보의 디지털 표현 원리를 설명할 수 있다.	효율적인 디지털 표현	동일한 정보가 다양한 방법으로 디지털로 변환되어 표현될 수 있음을 이해하고 정보 활용 목적에 따라 보다 효율적인 방법을 선택한다.
	정보의 표현 다양한 유형의 정보를 디지털로 표현할 수 있다.		컴퓨팅 환경에서 생산되는 방대하고 복잡한 종류의 자료들을 수집, 분석, 활용하기 위한 컴퓨팅 기술의 역할과 중요성을 이해한다.
정보의 관리	정보관리의 개념을 알고 설명할 수 있다.	자료의 분석/정보의 관리	인터넷, 응용 소프트웨어 등 컴퓨팅 도구를 활용하여 문제 해결을 위한 자료를 수집하고 분석한다.
	정보를 효율적으로 관리할 수 있다.		정보를 관리하는 데 적합한 컴퓨팅 도구를 선택하고 이를 활용하여 정보를 체계적으로 관리한다.

운영지침 ‘정보의 관리’ 내용 요소에 ‘정보관리의 개념을 알고 설명할 수 있다’, ‘정보를 효율적으로 관리할 수 있다’고 제시된 성취 기준이 ‘컴퓨팅 환경에서 생산되는 방대하고 복잡한 종류의 자료들을 수집, 분석, 활용하기 위한 컴퓨팅 기술의 역할과 중요성을 이해한다’, ‘인터넷, 응용 소프트웨어 등 컴퓨팅 도구를 활용하여 문제 해결을 위한 자료를 수집하고 분석한다’, ‘정보를 관리하는 데 적합한 컴퓨팅 도구를 선택하고 이를 활용하여 정보를 체계적으로 관리한다’ 로 보다 구체적으로 표현되어 자료의 분석과 정보의 관리에 있어 ‘컴퓨팅’을 강조한 것이라 파악된다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
문제의 구조화	문제 구조화의 개념과 방법을 설명할 수 있다. 주어진 문제를 해결하기 위해 구조화할 수 있다.	알고리즘 설계/ 알고리즘	복잡한 문제 상황에서 문제의 현재 상태, 목표 상태를 이해하고 목표 상태에 도달하기 위해 수행해야 할 작업을 분석한다.
문제의 추상화	문제 추상화의 개념과 방법을 설명할 수 있다. 주어진 문제를 추상화하여 표현할 수 있다.		복잡한 문제 상황에서 문제 해결에 불필요한 요소를 제거하거나 필요한 요소를 추출한다.
모델링과 시뮬레이션	모델링과 시뮬레이션의 개념을 설명할 수 있다. 실생활의 문제를 모델링할 수 있다. 실생활의 문제를 시뮬레이션할 수 있다.	분석	복잡하고 어려운 문제를 해결 가능한 작은 단위의 문제로 분해하고 모델링 한다.
복잡한 구조의 알고리즘 설계	선형구조, 비선형구조의 개념을 이해하고 알고리즘을 설계할 수 있다. 자료를 정렬하는 다양한 방법들을 알고 알고리즘을 설계할 수 있다. 자료를 탐색하는 다양한 방법들을 알고 알고리즘을 설계할 수 있다. 자료를 압축하는 다양한 방법들을 알고 알고리즘을 설계할 수 있다.	알고리즘 설계/ 알고리즘 분석	순차 구조, 선택 구조, 반복 구조 등의 제어 구조를 활용하여 논리적이고 효율적인 알고리즘을 설계한다.
알고리즘의 분석과 평가	같은 문제를 해결하는 데 사용되는 여러 알고리즘을 비교·평가할 수 있다.		다양한 알고리즘의 성능을 수행시간의 관점에서 분석하고 비교한다.

SW교육 운영지침		2015 개정 교육과정	
내용 요소	성취기준	내용 요소	성취기준
프로그래밍 언어의 분류	절차적 언어와 객체 지향적 언어의 특성을 설명할 수 있다.	프로그래밍 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.	
	프로그래밍 언어의 실행 절차를 설명할 수 있다.		
프로그래밍의 실제	변수와 상수를 이해하고, 이를 이용하여 자료의 입출력 프로그램을 작성할 수 있다.	변수와 자료형	자료형에 적합한 변수를 정의하고 이를 활용한 프로그램을 작성한다.
	조건문과 반복문을 이용하여 프로그램을 작성할 수 있다.	표준입출력 파일입출력	표준입출력과 파일입출력을 활용한 프로그램을 작성한다.
	다중 선택 구조를 이해하고 프로그램을 작성할 수 있다.	연산자	다양한 연산자를 활용한 프로그램을 작성한다.
	다중 반복 구조를 이해하고 프로그램을 작성할 수 있다.		순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.
프로그래밍의 실제	복잡한 제어 구조를 이용하여 프로그램을 작성할 수 있다.	중첩 제어 구조	중첩 제어 구조를 활용한 프로그램을 작성한다.
	배열의 개념과 필요성을 설명할 수 있다.	배열	배열의 개념을 이해하고 배열을 활용한 프로그램을 작성한다.
	배열을 이용하여 프로그램을 작성할 수 있다.		
	함수의 개념과 필요성을 설명할 수 있다.	함수	함수의 개념을 이해하고 함수를 활용한 프로그램을 작성한다.
함수와 변수의 참조 범위에 대해 설명할 수 있다.			
프로그래밍과 융합	함수를 활용하여 프로그램을 작성할 수 있다.		
	인문, 사회, 과학 분야에서 프로그래밍이 사용되는 사례를 찾아볼 수 있다.	프로그래밍 응용	다양한 학문 분야의 문제 해결을 위해 설계한 알고리즘을 프로그램으로 구현하고 효율성을 비교·분석한다.
	다양한 분야에서 문제해결과정을 찾아 이를 논리적으로 설계하고, 프로그램으로 작성할 수 있다.		
팀프로젝트 제작과 평가	팀을 이루어 하나의 프로젝트를 설계하고 제작할 수 있다.		
	팀에서 작성한 프로그램을 발표하고 평가할 수 있다.		다양한 학문 분야의 문제 해결을 위한 알고리즘을 협력하여 설계한다.

프로그래밍 영역에는 ‘표준입출력과 파일입출력’이라든가 ‘중첩’ 제어 구조 등으로 성취기준을 보다 명확하게 표현함으로써 교수와 학습 부담을 완화하고자 한 것으로 보인다.

수탁연구 CR 2015-35

SW교육 교수학습 모델 개발 연구

발행일 : 2015년 12월

발행인 : 백순근

발행처 : 한국교육개발원

주 소 : (06762) 서울시 서초구 바우뫼로 1길 35(우면동)

전화 02)3460-0114

FAX 02)3460-0121

<http://www.kedi.re.kr>

등 록 : 1973. 6. 13. 제16-35호

인 쇄 : 기쁜 D&P

전화 1800-2173

본 내용의 무단 복제를 금함.