



# 문제 해결과 프로그래밍

## 핵심 역량

- 1 정보 문화 소양 - 정보 윤리 인식, 정보 보호 능력, 정보 기술 활용 능력
- 2 컴퓨팅 사고력 - 추상화 능력, 자동화 능력, 창의 융합 능력
- 3 협력적 문제 해결력 - 협력적 컴퓨팅 사고력, 디지털 의사소통 능력, 공유와 협업 능력

## 대단원 지도 계획

중단원	소단원	차시	교과서 쪽수	학습 주제
Ⅲ- 1장 추상화	1-1 문제의 이해	13	p. 78~p. 81	• 문제 상황 파악 • 문제 분석과 분해
	1-2 핵심 요소 추출	14	p. 82~p. 88	• 문제의 패턴과 핵심 요소 • 추상화
Ⅲ- 2장 알고리즘	2-1 알고리즘의 이해	15~16	p. 92~p. 94	• 알고리즘의 개념 및 특징 • 알고리즘의 중요성
	2-2 알고리즘의 표현	17~18	p. 95~p. 106	• 알고리즘 표현 방법 • 알고리즘 설계 • 알고리즘의 활용

## 대단원 개관

이 단원에서는 실생활의 문제를 분석하고 핵심 요소를 추출하여 해결 가능한 형태로 만드는 추상화 과정을 경험한다. 그리고 문제 해결을 위한 방법과 절차인 알고리즘의 의미와 중요성에 대한 이해를 바탕으로 다양한 해법을 탐색하고 명확하게 표현하고자 하였다. 이러한 추상화와 알고리즘을 통해 설계한 문제 해결 과정을 자동화하는 능력을 함양하는 데 중점을 둔다.

## 대단원 학습 목표

- 실생활의 다양한 문제 상황을 분석하여 문제의 현재 상태와 목표 상태를 명확히 알고, 현재 상태에서 목표 상태에 도달하기 위해 수행할 작업의 종류와 순서를 이해한다.
- 문제 상황에 제시된 다양한 요소를 분석하여 목표 상태에 도달하기 위해 필수적인 요소를 찾을 수 있다.
- 알고리즘의 중요성을 인식하고 실생활에서 발생하는 문제의 해결 과정을 알고리즘으로 구상하여 논리적인 문제 해결 방법을 탐색할 수 있다.
- 다양한 형태의 자료를 외부로부터 입력받아 처리한 후 출력하는 프로그램을 작성할 수 있다.
- 변수의 필요성과 역할을 이해하고 변수의 값을 활용할 수 있다.
- 순차·선택·반복 구조의 명령 실행 과정을 이해하고 제어 구조를 이용해 효율적인 프로그램을 작성할 수 있다.
- 실생활의 다양한 문제 해결을 위한 소프트웨어를 협력적 프로젝트 수행을 통해 설계하고 개발할 수 있다.

## Ⅲ- 3장 프로그래밍

3-1 프로그래밍 3-2 입력과 출력	19~20	p. 110~p. 112 p. 113~p. 116	• 프로그래밍 이해하기 • 입력과 출력 이해하기 • 입력과 출력 실습하기
3-3 변수와 연산	21~22	p. 117~p. 121	• 변수 학습하기 • 연산자 학습하기 • 프로그램 심화시키기
3-4 제어 구조	23~24	p. 122~p. 124	• 순차 구조 이해하기 • 반복 구조 이해하기 • 선택 구조 이해하기
3-5 프로그래밍 응용	25	p. 125~p. 126	• 프로젝트 1: '사과 받기' 게임 재구성하기
	26	p. 127~p. 128	• 프로젝트 2: 친구와 이야기 속으로
	27	p. 129~p. 131	• 프로젝트 3: 스마트 선풍기
	28	p. 132~p. 134	• 프로젝트 4: 자동 문자 응답 시뮬레이터

# 13차시

소단원 지도 계획

## 1-1 문제의 이해

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 1 추상화

핵심 역량 ▶ 추상화 능력

수업 시간 ▶ 1시간

교과서 쪽수 ▶ 78~81쪽

학습 목표 ▶ 실생활 문제의 목표 상태를 파악하고 목표 상태 도달을 위한 문제 분석과 분해를 할 수 있다.

수업 방법 ▶ 강의, 토론, 발표

기능 ▶ 비교하기, 분석하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	• 설록 홈즈의 추추는 암호 만화를 보고 의미 전달을 위한 다양한 방법을 생각해 본다.	사진 자료 영상 자료	• 만화에 나오는 간단한 암호를 풀어 보고 문자를 간단한 그림으로도 표현할 수 있음을 경험하게 한다.
전개	<b>1 문제 상황 파악</b> <ul style="list-style-type: none"> <li>문제 해결을 위한 현재 상태와 목표 상태의 개념을 이해한다.</li> <li>숫자 조각 퍼즐을 보며 현재 상태와 목표 상태를 이해한다.</li> <li>[함께 해 보기]의 은아의 방 상황에 대한 글과 그림을 살펴보고 현재 상태와 목표 상태를 알아본다.</li> <li>[함께 해 보기]의 컨테이너 옮기기 문제의 현재 상태와 목표 상태를 파악하고 문제 해결 과정을 생각해 본다.</li> </ul>	사진 자료 교과서 활동지	• 다양한 상황에서의 현재 상태와 목표 상태를 찾아보게 한다.
	<b>2 문제 분석과 분해</b> <ul style="list-style-type: none"> <li>은아네 거실 상황을 파악하는 활동을 통해 문제 해결 과정에서 문제를 정확하게 이해하고 분석하는 것의 중요성을 이해한다.</li> <li>문제 해결을 위한 처음 전략으로 분해 과정의 필요성을 이해한다.</li> </ul>	사진 자료 교과서 활동지	• 문제 분해는 0과 1로만 처리하는 컴퓨터의 특성에 의한 필수 과정임을 염두에 두고 분해 과정을 경험하도록 한다.
	<ul style="list-style-type: none"> <li>[함께 해 보기]의 '청소 문제 분해' 활동을 통해 상황을 다양한 방식으로 분해해 본다.</li> <li>[함께 해 보기]의 '소프트웨어 모듈 분해' 활동을 통해 다양한 기능을 생각하여 분해해 본다.</li> <li>[함께 해 보기]의 '자동차 번호판 인식을 위한 분해' 활동을 통해 패턴 인식을 위한 분해 활동을 해 본다.</li> </ul>	사진 자료 교과서 활동지	• 다양한 접근의 문제 분해 과정을 경험하게 한다.
정리	• 일상생활에서 문제를 분해하여 컴퓨터로 해결하면 좋은 사례를 친구들과 찾아본다.	정리 및 평가 활동지	• 학교, 교실, 집, 마트 등의 주변에서 생각해 보도록 한다.
평가	<ul style="list-style-type: none"> <li>문제 분석 과정을 통해 문제 상황의 현재 상태와 목표 상태를 파악하는가?</li> <li>목표 상태에 도달하기 위해 문제 상황을 세부적으로 분해할 수 있는가?</li> </ul>	수행 평가지 관찰 평가지	

# 14차시

소단원 지도 계획

## 1-2 핵심 요소 추출

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 1 추상화

핵심 역량 ▶ 추상화 능력, 창의 융합 능력

수업 시간 ▶ 1시간

교과서 쪽수 ▶ 82~88쪽

학습 목표 ▶ 문제 상황의 패턴을 분석하여 핵심 요소를 찾아 다양한 방식으로 추상화할 수 있다.

수업 방법 ▶ 강의, 발표

기능 ▶ 분석하기, 핵심 요소 추출하기, 표현하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	• 컴퓨터로 문제를 해결하기 위해 추상화 과정이 왜 필요한 지에 대해 이야기해 본다.		
전개	<b>1 문제의 패턴과 핵심 요소</b> <ul style="list-style-type: none"> <li>다양한 패턴의 예를 보며 패턴과 패턴 인식의 개념을 이해한다.</li> <li>[함께 해 보기]의 기호, 이미지, 동작 등의 다양한 패턴 찾기 활동을 해 본다.</li> </ul>	사진 자료 교과서 활동지	• 다양한 사례에서 규칙성을 찾아봄으로써 패턴을 이해하도록 한다.
	<ul style="list-style-type: none"> <li>핵심 요소의 개념을 이해하고 추출하는 방법을 알아본다.</li> <li>[함께 해 보기]를 통해 다양한 문제 상황에서의 핵심 요소를 추출해 본다.</li> </ul>	사진 자료 교과서 활동지	<ul style="list-style-type: none"> <li>핵심 요소를 파악하기 전에 문제를 작게 분해했는지 확인하도록 한다.</li> <li>불필요한 요소와 필요한 요소를 구분하는 기준에 대해 토의해 보면서 핵심 요소의 개념을 이해한다.</li> </ul>
	<b>2 추상화</b> <ul style="list-style-type: none"> <li>추상화의 개념과 필요성을 이해하고 다양한 추상화의 방법을 이해한다.</li> <li>[함께 해 보기]를 통해 추상화의 다양한 방법을 연습한다.</li> </ul>	교과서 활동지	<ul style="list-style-type: none"> <li>핵심 요소를 이용하여 규칙성을 다양한 방법으로 표현해 보도록 한다.</li> <li>컴퓨터로 처리하기 위해 추상화 과정이 왜 필요한지 0과 1로 처리하는 컴퓨터의 특징과 관련지어 설명한다.</li> </ul>
	<b>2 수식화, 도식화 모델 정의</b> <ul style="list-style-type: none"> <li>추상화의 다양한 표현 방법을 알아본다.</li> <li>만화를 보며 추상화의 필요성을 생각해 보도록 한다.</li> <li>[함께 해 보기]를 통해 다양한 추상화 표현 방법을 경험해 본다.</li> </ul>	교과서 활동지	• 문제의 특성에 맞는 추상화 표현 방법을 생각해 보도록 한다.
정리	• [실습 활동]을 통해 학습 내용을 정리해 본다.	교과서 활동지	• [1-1 문제 이해] 단원과 [1-2 핵심 요소 추출] 단원의 핵심 내용을 모두 연결해서 이해하도록 한다.
평가	<ul style="list-style-type: none"> <li>문제 해결을 위한 핵심 요소를 구분하였는가?</li> <li>핵심 요소와 패턴을 이해하고 문제 상황을 추상화할 수 있는가?</li> </ul>	수행 평가지 자기 평가지 관찰 평가지	

## 2-1 알고리즘 이해

대단원 ▶ III 문제 해결과 알고리즘

중단원 ▶ 2 알고리즘

핵심 역량 ▶ 추상화·창의 융합 능력, 협력적 컴퓨팅 사고력

수업 시간 ▶ 2시간

교과서 쪽수 ▶ 92~94쪽

학습 목표 ▶ 사람과 컴퓨터의 문제 해결 방법의 차이를 설명할 수 있다.  
알고리즘의 개념을 이해하고 중요성을 인식할 수 있다.

수업 방법 ▶ 강의, 조사, 토론, 발표, 프로젝트

기능 ▶ 탐색하기, 분석하기, 실천하기, 계획하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>교과서 여쪽 동기 유발 만화 자료를 통해 위대한 아이디어를 구체화시킨 알고리즘 적용 사례에 대해 이야기해 보고 이 외에 다른 사례가 있는지 발표해 본다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>		<ul style="list-style-type: none"> <li>위대한 아이디어를 구체화시킨 알고리즘 사례에 대해 소개한다.</li> </ul>
전개	<b>1</b> 알고리즘의 개념 및 조건 <ul style="list-style-type: none"> <li>알고리즘의 개념을 정확하게 이해하고 효율적인 알고리즘을 작성하기 위해서는 어떤 조건이 필요한지 예를 들어 설명한다.</li> <li>여러 개의 숫자 카드 중에서 임의의 숫자를 찾아내는 방법에 대해 의견을 나누고 발표해 본다.</li> </ul>	교과서 그림 자료 활동지	<ul style="list-style-type: none"> <li>[함께 해 보기] 활동은 모둠별 과제로 제시해 주어 조사 활동을 통해 발표하게 하는 것도 좋다.</li> </ul>
	<b>2</b> 알고리즘의 효율성 <ul style="list-style-type: none"> <li>효율적인 알고리즘 설계의 중요성을 이해한다.</li> <li>알고리즘 설계에 따라 비용과 효율성에 많은 차이가 생긴다는 것을 이해하도록 돕는다.</li> <li>정렬되어 있는 숫자 카드를 찾아내는 두 가지 방법을 비교하여 효율적인 알고리즘이 무엇인지 알아본다.</li> <li>그림 자료를 보며 정렬되어 있는 숫자 카드를 찾는 문제의 해결 방법을 같이 고민해 본다.</li> <li>목적지까지 이동하려는 로봇을 통해 알고리즘의 중요성을 알아본다.</li> <li>[함께 해 보기] 활동으로 목적지까지 이동하려는 로봇을 알고리즘으로 설계하여 친구들과 비교해 본 후 효율적인 알고리즘은 무엇인지 의견을 나누어 보도록 한다.</li> </ul>	교과서 그림 자료 활동지	
정리	<ul style="list-style-type: none"> <li>알고리즘의 개념과 조건, 효율적인 알고리즘 설계의 중요성에 대해 다시 한 번 확인하며 정리한다.</li> </ul>		
평가	<ul style="list-style-type: none"> <li>알고리즘의 개념은 무엇이며, 알고리즘을 작성하기 위해서는 어떤 조건이 필요한가?</li> <li>효율적인 알고리즘을 설계하는 것은 왜 중요한가?</li> </ul>	자기 평가지 동료 평가지 관찰 평가지	

## 2-2 알고리즘의 표현

대단원 ▶ III 문제 해결과 알고리즘

중단원 ▶ 2 알고리즘

핵심 역량 ▶ 추상화·창의 융합 능력, 협력적 컴퓨팅 사고력

수업 시간 ▶ 2시간

교과서 쪽수 ▶ 95~106쪽

학습 목표 ▶ 알고리즘을 다양한 방법으로 표현하고 설계할 수 있다.

수업 방법 ▶ 강의, 조사, 토론, 발표, 프로젝트

기능 ▶ 탐색하기, 분석하기, 실천하기, 계획하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>음악을 표현하기 위해 악보를 사용하는 것처럼 문제 해결 절차를 표현하기 위해 어떤 방법을 사용하는지 이야기해 보고 발표한다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 그림 자료	<ul style="list-style-type: none"> <li>우리의 생활 주변에서 효과적인 문제 해결을 위해 약속된 기호나 표현 방법에 대해 소개한다.</li> </ul>
전개	<b>1</b> 알고리즘 표현 방법 <ul style="list-style-type: none"> <li>자연어, 의사 코드, 순서도를 활용한 알고리즘 표현 방법을 장단점, 특징 위주로 설명한다.</li> <li>자판기 동작 원리와 짝수·홀수를 판별하는 예제를 통해 자연어, 의사 코드, 순서도 표현 방법을 이해한다.</li> </ul>	교과서 그림 자료 활동지	<ul style="list-style-type: none"> <li>[함께 해 보기] 활동은 모둠별 과제로 제시해 주어 조사 활동을 통해 발표하게 하는 것도 좋다.</li> </ul>
	<b>2</b> 알고리즘의 설계 <ul style="list-style-type: none"> <li>알고리즘의 설계 구조를 순차 구조, 선택 구조, 반복 구조로 나누어 표현하는 방법과 특징을 설명한다.</li> <li>순차 구조에서 여러 번 반복되는 동작을 반복 구조로 표현하면 간편하고 효율적으로 표현할 수 있음을 이해한다.</li> <li>그림 자료를 보며 반복적인 동작의 표현 방법을 같이 고민해 본다.</li> <li>[함께 해 보기] 활동으로 선택 구조, 반복 구조를 이용하여 효율적으로 알고리즘을 표현해 보도록 한다.</li> </ul>	교과서 그림 자료 활동지	
	<b>3</b> 알고리즘의 활용 <ul style="list-style-type: none"> <li>정렬과 탐색 알고리즘 등 실생활의 다양한 분야에서 활용되고 있는 알고리즘 활용 사례를 같이 고민해 보고 의견을 나누도록 한다.</li> </ul>	교과서 그림 자료 활동지	
정리	<ul style="list-style-type: none"> <li>알고리즘의 표현 방법, 설계 구조, 활용 분야에 대해 다시 한 번 확인하며 정리한다.</li> </ul>		
평가	<ul style="list-style-type: none"> <li>알고리즘은 어떤 방법으로 표현하는가?</li> <li>알고리즘의 설계 구조에는 어떤 것이 있는가?</li> <li>정렬, 탐색 알고리즘을 이해하고 있는가?</li> </ul>	자기 평가지 동료 평가지 관찰 평가지	

## 3-1 프로그래밍, 3-2 입력과 출력

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅 사고력, 공유와 협업 능력

수업 시간 ▶ 2시간

교과서 쪽수 ▶ 110~112쪽, 113~116쪽

학습 목표 ▶ 프로그래밍에 대해 이해하고 프로그래밍 언어를 이용하여 자료의 입력과 출력을 할 수 있다.

수업 방법 ▶ 강의, 조사, 토론, 발표, 실습

기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>주변에서 사용하는 애플리케이션을 비롯한 각종 프로그램에는 무엇이 있는지 이야기해 본다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 111쪽 이야기 자료	<ul style="list-style-type: none"> <li>동기 유발 단계에서 각종 프로그램에 프로그래밍 언어가 존재한다는 사실을 이해할 수 있도록 이끈다.</li> </ul>
전개	<b>1 프로그래밍 이해하기</b> <ul style="list-style-type: none"> <li>프로그래밍의 정의에 대해 배워 보고 프로그래밍과 관련된 여러 가지 용어에 대해 익힌다.</li> <li>프로그래밍 과정에서 사용되는 프로그래밍 언어의 유형과 교육용 프로그래밍 언어에 대해 알아본다.</li> <li>[함께 해 보기]를 통해 프로그래밍 과정 단계와 다양한 프로그래밍 언어를 알아본다.</li> </ul>	교과서 112쪽 이야기 자료	<ul style="list-style-type: none"> <li>프로그래밍과 관련된 여러 가지 용어를 구분하여 이해할 수 있게 한다. 또한 다양한 고급 언어를 소개한다.</li> <li>일반 프로그래밍 언어와 교육용 프로그래밍 언어의 인터페이스를 비교할 수 있도록 자료를 제시하거나 교육용 프로그래밍 언어를 체험해 볼 수 있는 시간을 가진다.</li> <li>프로그래밍의 패러다임 변화를 소개할 수 있다.</li> </ul>
	<b>2 입력과 출력 이해하기</b> <ul style="list-style-type: none"> <li>지도 검색 프로그램에서 길 찾기할 때를 생각하며 컴퓨터의 입출력을 이해한다.</li> <li>스크래치 프로그램을 살펴보고 프로그래밍 언어의 입력과 출력에 대해 이해한다.</li> </ul>	교과서 113쪽 이야기 자료, 지도 검색 프로그램, 스크래치	<ul style="list-style-type: none"> <li>실제로 지도 검색 프로그램에서 길 찾기를 하면서 입력과 출력에 대해 알아본다.</li> <li>사람의 정보 처리 과정과 컴퓨터의 입출력을 비교해 본다.</li> </ul>
	<b>3 입력과 출력 실습 활동</b> <ul style="list-style-type: none"> <li>핵심 명령 블록과 따라하기를 보며 키보드를 통해 자료를 입력하고 출력한다.</li> <li>핵심 명령 블록과 따라하기를 보며 마우스를 통해 자료를 입력하고 출력한다.</li> </ul>	스크래치 카드, 스크래치	<ul style="list-style-type: none"> <li>핵심 명령 블록의 카테고리별로 교과서에 제시된 것 외에 다른 것들도 다양하게 사용해 볼 수 있게 한다.</li> <li>기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 본다.</li> </ul>
정리	<ul style="list-style-type: none"> <li>프로그래밍의 용어와 과정에 대해 발표하고 실습한 것을 공유하며 프로그래밍에서의 입력과 출력에 대해 다시 떠올려본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>사용할 프로그래밍 언어의 개발 환경 및 특성을 이해하고 있는가?</li> <li>프로그래밍의 과정을 이해하고 있는가?</li> <li>프로그래밍에서 자료의 입력과 출력을 할 수 있는가?</li> </ul>	자기 평가지 동료 평가지	

## 3-3 변수와 연산

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅 사고력, 공유와 협업 능력

수업 시간 ▶ 2시간

교과서 쪽수 ▶ 117~121쪽

학습 목표 ▶ 변수의 개념을 이해하고 변수와 연산자를 활용한 프로그램을 작성할 수 있다.

수업 방법 ▶ 강의, 조사, 토론, 발표, 실습

기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>온도계와 속도계에 대해 이야기하며 프로그래밍에서 수시로 변하는 수치를 어떻게 처리하고 연산할지 예상해 본다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 117쪽 이야기 자료	<ul style="list-style-type: none"> <li>우리 주변에서 수시로 변하는 수치의 예를 다양하게 들 수 있다.</li> </ul>
전개	<b>1 변수 학습하기</b> <ul style="list-style-type: none"> <li>프로그래밍에서 변수란 무엇이고 변수의 유형에는 무엇이 있는지 살펴본다.</li> <li>변수를 만들 때 변수의 초기화에 대해 알아보고 따라하기와 핵심 명령 블록을 통해 변수를 만드는 방법을 알아본다.</li> </ul>	교과서 119쪽 스크래치 카드, 스크래치 CD 자료	<ul style="list-style-type: none"> <li>변수의 기능과 역할에 대해 소개한다. 변수와 변수 값의 의미를 이해할 수 있도록 지도한다.</li> <li>기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 본다.</li> </ul>
	<b>2 연산자 학습하기</b> <ul style="list-style-type: none"> <li>프로그래밍에서 연산자란 무엇이고 연산자의 유형에는 무엇이 있는지 살펴본다.</li> <li>핵심 명령 블록과 [따라 하기]를 보며 연산자의 종류를 이해하고 활용해 본다.</li> <li>[따라 하기]와 [함께 해 보기]를 통해 연산자를 활용한 프로그램을 만들어 본다.</li> </ul>		<ul style="list-style-type: none"> <li>기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 본다.</li> </ul>
	<b>3 프로그램 심화시키기</b> <ul style="list-style-type: none"> <li>50보다 작고 큰 수를 비교하는 스크립트를 수정하여 짝수와 홀수를 판단하는 프로그램을 작성해 본다.</li> </ul>		<ul style="list-style-type: none"> <li>이 외에도 지도서 참고 자료를 활용하여 다양한 심화 학습을 진행한다.</li> </ul>
정리	<ul style="list-style-type: none"> <li>프로그래밍에서 변수와 연산자에 대해 발표하고 실습한 것을 공유하며 변수와 연산자에 대해 다시 떠올려 본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>프로그래밍에서 변수를 만들어 활용할 수 있는가?</li> <li>프로그래밍에서 연산자를 활용할 수 있는가?</li> </ul>	자기 평가지 동료 평가지	



## 3-4 제어 구조

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅  
사고력, 공유와 협업 능력

수업 시간 ▶ 2시간

교과서 쪽수 ▶ 122~124쪽

학습 목표 ▶ 순차, 반복, 선택의 개념과 원리를 이해하고 세 가지 구조를 활용한 프로그램을 작성할 수 있다.

수업 방법 ▶ 강의, 조사, 토론, 발표, 실습

기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>일상생활에서 일을 처리할 때를 떠올리며 프로그램을 구현할 때 필요한 절차 구조에 대해 예상한다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 122쪽 이야기 자료	<ul style="list-style-type: none"> <li>효율적인 프로그램을 작성하기 위해서는 제어 구조에 대해 생각해야 된다는 것을 이해하도록 한다.</li> </ul>
전개	<b>1</b> 순차 구조 이해하기 <ul style="list-style-type: none"> <li>일상생활과 비교하여 순차 구조에 대해 이해한다.</li> <li>핵심 명령 블록과 [따라 하기]를 보며 순차 구조를 활용해 본다.</li> </ul>	교과서 123쪽 스크래치 카드, 스크래치	<ul style="list-style-type: none"> <li>기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다.</li> </ul>
	<b>2</b> 반복 구조 이해하기 <ul style="list-style-type: none"> <li>프로그램을 작성할 때 중복되는 부분을 효율적으로 처리하기 위해 반복 구조를 실행하며 조건 영역과 실행 영역으로 구분됨을 이해한다.</li> <li>핵심 명령 블록과 [따라 하기], [함께 해 보기]를 보며 반복 구조를 활용해 본다.</li> </ul>		<ul style="list-style-type: none"> <li>기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다.</li> </ul>
	<b>3</b> 선택 구조 이해하기 <ul style="list-style-type: none"> <li>선택 구조의 뜻을 이해하고 조건 명령의 조건식에 따라 실행할 명령들에 대해 알아본다.</li> <li>핵심 명령 블록과 [따라 하기], [함께 해 보기]를 보며 선택 구조를 활용해 본다.</li> </ul>		<ul style="list-style-type: none"> <li>기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다.</li> <li>10블록 평가 후에는 디버깅과 테스트를 해 본다.</li> </ul>
정리	<ul style="list-style-type: none"> <li>프로그래밍에서 순차, 반복, 선택 구조에 대해 발표하고 실습한 것을 공유하며 제어 구조를 다시 떠올려 본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>프로그래밍에서 순차 구조를 활용할 수 있는가?</li> <li>프로그래밍에서 반복 구조를 활용할 수 있는가?</li> <li>프로그래밍에서 선택 구조를 활용할 수 있는가?</li> </ul>	디자인 저널	

## 3-5 프로젝트 1: '사과 받기' 게임 재구성

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅  
사고력, 공유와 협업 능력

수업 시간 ▶ 1시간

교과서 쪽수 ▶ 125~126쪽

학습 목표 ▶ 주어진 프로젝트를 탐색, 수정해 보는 활동을 통해 컴퓨팅 사고를 기를 수 있다.

수업 방법 ▶ U-M-C 전략(재구성 중심 모형)

기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>게임을 하면서 캐릭터나 기능 등을 바꾸어 보고 싶었던 경험에 대해 이야기를 나눈다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	컴퓨터실 스크래치	<ul style="list-style-type: none"> <li><a href="http://scratch.mit.edu">http://scratch.mit.edu</a> 또는 scratch 2.0</li> </ul>
전개	<b>1</b> 놀이하기(Using) <ul style="list-style-type: none"> <li>'사과 받기' 게임 놀이를 하며 게임 속의 구성 요소와 기능에 대해 탐색해 본다.</li> <li>꼭 '사과 받기' 게임이 아니어도 무방하므로 학생들이 미리 자신이 바꾸어 보고 싶은 게임의 프로젝트를 준비해 와도 좋다.</li> </ul>	'사과 받기' 게임 프로젝트 파일 CD 자료  교과서 125쪽 [게임을 바꿔 보자] 활동	<ul style="list-style-type: none"> <li>CD 자료에 있는 '사과 받기' 게임 프로젝트 파일을 미리 온라인 사이트에 탑재하여 학생들이 수업 시간에 바로 다운로드할 수 있도록 한다.</li> <li>게임을 실행해보기에 앞서 먼저 게임을 하면서 무엇을 바꾸고 싶은지에 대해 생각하면서 놀이해 보라고 안내한다.</li> <li>학생들의 수준에 따라 재구성의 난이도를 조절할 수 있다(수준이 낮은 경우에는 구성 요소의 간단한 바꾸기, 수준이 높은 경우에는 기능의 다양한 변화를 통해 게임의 레벨을 몇 단계로 바꿀 수도 있으며, 그 과정에서 다양한 명령어의 기능을 익히고 적용해 볼 수 있다).</li> </ul>
	<b>2</b> 수정하기(Modify) <ul style="list-style-type: none"> <li>'사과 받기' 게임을 구성 요소(사과, 바구니)를 다른 객체로 바꾸거나 장면 등을 바꾸어 보도록 한다.</li> <li>구성 요소의 기능과 게임의 규칙 등을 명령어의 수정, 복제, 추가, 확장 등과 같은 방법을 통해 바꾸어 보도록 한다.</li> </ul>		
	<b>3</b> 재구성하기(reCreate) <ul style="list-style-type: none"> <li>학습지에 작성하였던 계획들을 실제 코딩을 하며 게임을 재구성하여 자신만의 새로운 게임으로 만들어 본다.</li> <li>테스팅과 디버깅 과정을 통해 프로그램을 개선한다.</li> </ul>		
정리	<ul style="list-style-type: none"> <li>재구성한 게임을 친구들과 함께 발표하고 공유하며, 똑같은 프로젝트가 어떻게 다양하게 변화하였는지를 살펴본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>구성 요소를 다양하게 변화시켰는가?</li> <li>명령어의 수정, 복제, 추가, 확장 등의 재구성이 오류 없이 잘 실행되는가?</li> <li>나만의 생각이 실제 프로그램에 잘 구현이 되었는가?</li> </ul>	자기 평가지 동료 평가지	

## 3-5 프로젝트 2: 친구와 이야기 속으로

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅  
사고력, 공유와 협업 능력

수업 시간 ▶ 1시간

교과서 쪽수 ▶ 127~128쪽

학습 목표 ▶ 이야기를 만드는 활동을 통해 배경과 객체 사이의 상호 작용을 이해하며 컴퓨팅 사고력을 기를 수 있다.

수업 방법 ▶ 짝 프로그래밍

기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>‘어린 왕자’ 이야기를 떠올리며 전체적인 줄거리에 대해 이야기를 나눈다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 127쪽 이야기 자료	<ul style="list-style-type: none"> <li>‘어린 왕자’가 아닌 다른 이야기를 선택하여도 무방하다.</li> </ul>
전개	<p><b>1</b> 이야기 재구성하기</p> <ul style="list-style-type: none"> <li>두 명이 한 팀을 이룬다.</li> <li>어린 왕자의 이야기 중에서 컴퓨터로 구현하고 싶은 장면 3~4개를 선택한다.</li> <li>컴퓨터로 구현할 수 있도록 이야기를 재구성하여 간단히 그려 본다.</li> </ul> <p><b>2</b> 객체 선정하기</p> <ul style="list-style-type: none"> <li>각 장면에서 이야기를 구성할 캐릭터, 멀티미디어 자료, 애니메이션 요소, 방송의 흐름 등을 선정하여 정리 한다.</li> </ul> <p><b>3</b> 알고리즘 설계하기</p> <ul style="list-style-type: none"> <li>사용된 객체의 주요 기능을 알고리즘으로 설계해 본다.</li> </ul> <p><b>4</b> 개발하기(짝 프로그래밍)</p> <ul style="list-style-type: none"> <li>친구와 컴퓨터로 구현하기 위해 제한된 조건을 지키며 프로그래밍한다.</li> <li>한 명당 한 번에 5개의 블록만 사용할 수 있다.</li> <li>한 번의 기회에 3분을 초과할 수 없다.</li> <li>다른 사람이 만든 명령 블록을 수정, 삭제할 수 있다.</li> </ul>	<p>교과서 127쪽 [이야기 재구성하기] 활동 자료</p> <p>[객체 선정하기]</p> <p>[알고리즘 설계하기] 활동 자료</p>	<ul style="list-style-type: none"> <li>디자인 기반 학습 모형을 기반으로 학생들이 스스로 설계하고 재구성하며 반복적으로 공유하고 피드백하여 완성하도록 한다.</li> <li>이야기를 재구성하면서 전체 이야기에 대한 설계가 이루어진다.</li> <li>멀티미디어 자료 요소에서 학생들 모습이 담긴 사진(웹 카메라로 사진 찍기), 목소리를 직접 녹음하기 등 스크래치 기능을 활용하여 자료를 수집하여도 좋다.</li> <li>짝 프로그래밍에서 제시되는 조건을 학생들과 함께 논의하여 바꿀 수도 있다. 하지만 꼭 지켜야 한다는 약속을 프로그래밍 전에 해야 한다.</li> </ul>
정리	<ul style="list-style-type: none"> <li>짝과 함께 개발한 어린 왕자 스토리를 다른 친구들과 공유하여 잘된 점과 개선할 점을 알아본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>어린 왕자 이야기 만들기에서 배경과 객체 사이의 상호 작용을 효과적으로 나타내었는가?</li> <li>짝 프로그래밍 과정에서 짝과 협력하여 주어진 조건을 잘 지키며 프로그래밍하였는가?</li> </ul>	자기 평가지 동료 평가지	<ul style="list-style-type: none"> <li>짝 프로그래밍에서 중요하게 여기는 협력적 컴퓨팅이 잘 이루어졌는지 확인한다.</li> </ul>

## 3-5 프로젝트 3: 스마트 선풍기

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅  
사고력, 공유와 협업 능력

수업 시간 ▶ 1시간

교과서 쪽수 ▶ 129~131쪽

학습 목표 ▶ 일상생활의 문제 상황을 분석하여 설계와 구현을 통해 해결해 봄으로써 컴퓨팅 사고력을 기를 수 있다.

수업 방법 ▶ D-D-D 전략(개발 중심 모형)

기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>일상생활에서 개발과 구현을 통해 보다 편리한 생활을 하고 있는 예를 찾아본다.</li> <li>만화 내용을 읽고 문제 상황을 이해해 본다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 129쪽 이야기 자료	<ul style="list-style-type: none"> <li>만화를 보고 문제 상황을 이해하고 태양광을 충전하여 사용하는 상황들에 대해 이야기해 보며 공감대를 높인다.</li> </ul>
전개	<p><b>1</b> 탐구하기</p> <ul style="list-style-type: none"> <li>문제 상황에 대한 현재 상태와 목표 상태를 알아본다.</li> <li>태양열을 이용한 스마트 선풍기를 작동시키기 위한 기능의 분해 과정을 완성해 본다.</li> </ul> <p><b>2</b> 설계하기</p> <ul style="list-style-type: none"> <li>핵심 요소의 세부 조건들을 추상화해 본다.</li> <li>스마트 선풍기를 구현하기 위한 알고리즘을 완성해 본다.</li> </ul> <p><b>3</b> 구현하기</p> <ul style="list-style-type: none"> <li>3가지 핵심 요소에 대한 알고리즘을 이용하여 프로그램을 개발해 본다.</li> </ul>	교과서 129쪽 활동 자료 CD 자료	<ul style="list-style-type: none"> <li>DDD 모형을 적용하여 탐구와 설계 과정을 거쳐 개발하도록 한다.</li> <li>학생들이 스스로 문제를 파악하고 추상화와 알고리즘 과정의 빈칸을 채워 보도록 한다.</li> <li>혼자 해결하기 힘든 학생들은 둘 이상의 학생이 서로 협력하여 탐구하고 개발하도록 한다.</li> </ul>
정리	<ul style="list-style-type: none"> <li>되돌아보기를 통해 수정·보완해 본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>친구들과 공유하여 개선 방안을 찾았는가?</li> <li>개선해야 하는 부분을 디버깅하였는가?</li> </ul>	디자인 저널	

## 3-5 프로젝트 4: 자동 문자 응답 시뮬레이터

대단원 ▶ III 문제 해결과 프로그래밍

중단원 ▶ 3 프로그래밍

핵심 역량 ▶ 추상화, 자동화, 창의 융합, 협력적 컴퓨팅 사고력, 공유와 협업 능력

수업 시간 ▶ 1시간

교과서 쪽수 ▶ 132~134쪽

학습 목표 ▶ 일상생활의 문제 상황에 대한 요구 분석을 바탕으로 설계 및 구현하여 컴퓨팅 사고력을 기를 수 있다.

수업 방법 ▶ D-N-I-S 전략(디자인 중심 모형)

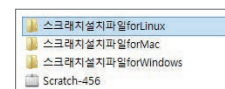
기능 ▶ 분석하기, 표현하기, 수집하기, 관리하기

단계	교수·학습 활동	학습 자료	지도상의 유의점
도입	<ul style="list-style-type: none"> <li>만화 내용을 읽고 문제 상황을 이해해 본다.</li> <li>학습 문제와 학습 활동을 확인한다.</li> </ul>	교과서 132쪽 이야기 자료	<ul style="list-style-type: none"> <li>만화를 보고 문제 상황을 이해하고 태양광을 충전하여 사용하는 상황들에 대해 이야기해보며 공감대를 높인다.</li> </ul>
전개	<p><b>1</b> 요구 분석하기</p> <ul style="list-style-type: none"> <li>글에서 제시된 문제 상황의 목표 상태를 적어 본다.</li> <li>수신 문자를 수집하고 어떤 문자에 답장을 보내야 하는지 수신 번호들의 패턴을 찾아보도록 한다.</li> <li>응답 문자를 보내야 하는 번호와 그렇지 않은 번호를 구분하는 기준을 이야기한다.</li> </ul> <p><b>2</b> 설계하기</p> <ul style="list-style-type: none"> <li>자동 문자 응답 시뮬레이터의 핵심 요소, 패턴, 추상화 과정을 완성해 본다.</li> <li>자동 문자 전송 앱의 작동 순서를 세부적으로 나누어 알고리즘을 채워 본다.</li> </ul> <p><b>3</b> 구현하기</p> <ul style="list-style-type: none"> <li>실행 화면과 주요 스크립트를 참고하여 자동 문자 응답 시뮬레이터를 제작해 본다.</li> </ul> <p><b>4</b> 공유하기</p> <ul style="list-style-type: none"> <li>다른 친구들과 공유하여 개선 방법을 논의해 본다.</li> </ul>	교과서 133쪽 활동 자료 CD 자료	<ul style="list-style-type: none"> <li>실제 학생들이 최근에 받은 문자들을 보며 부재중 응답 문자를 보내야 하는 문자와 그렇지 않은 문자들로 분류해 본다.</li> <li>핵심 요소별 기능에 대한 추상화와 알고리즘을 연결하여 논리적으로 일관성 있게 완성해 본다.</li> <li>혼자 해결하기 힘든 학생들은 둘 이상의 학생이 서로 협력하여 탐구하고 개발하도록 한다.</li> </ul>
정리	<ul style="list-style-type: none"> <li>되돌아보기를 통해 수정·보완해 본다.</li> <li>일상생활에서의 문제 상황을 이야기하며 프로그래밍 구현을 통해 개선할 수 있는 상황들에 대해 논의해 본다.</li> </ul>	디자인 저널	
평가	<ul style="list-style-type: none"> <li>사용자의 요구를 분석하여 설계 과정에 반영할 수 있는가?</li> <li>친구들과 공유하여 개선 방안을 디버깅할 수 있는가?</li> </ul>	자기 평가지 동료 평가지	

## 부록 CD의 스크래치 소스 파일 사용하는 법



- 스크래치 2.0 오프라인 에디터 다운로드 사이트(<https://scratch.mit.edu/scratch2download/>)에 가거나 부록 CD의 '설치 파일' 폴더를 연다.
- 부록 CD의 '스크래치 설치 파일' 폴더를 열면 운영 체제별로 설치 파일이 정리되어 있다.

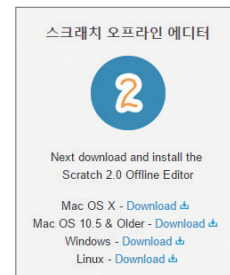


- 원도 환경인 경우 '스크래치설치파일forWindows' 폴더를 클릭한다. 운영 체제별로 어도비 에어와 설치 파일이 각각 들어 있다.

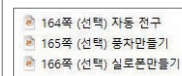
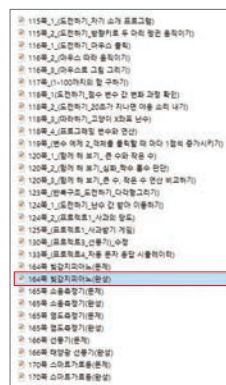
- 먼저 어도비 에어(Adobe AIR)를 설치한다. AdobeAIRinstaller.exe를 더블 클릭하면 된다.



- 스크래치 설치 파일을 설치한다. Scratch-456 .exe를 더블 클릭하면 스크래치가 설치되고 바탕 화면에 스크래치 설치 아이콘(Scratch 2)이 만들어 진다.



- '교과서 소스'와 '지도서 소스' 폴더 중 '교과서 소스' 폴더를 클릭하여 해당 페이지에 맞는 소스 파일을 선택한다.



▲ 교과서 소스

▲ 지도서 소스, 추가 프로젝트 소스가 들어있다.



‘셜록 홈즈의 춤추는 인형 암호’ 애플리케이션을 이용하여 학생들이 직접 암호를 만들어 볼 수 있다.

생각해 보기

- 간단한 그림이나 기호로 의미를 대신하고 있는 것-심벌 마크, 이모티콘, 아이콘, 암호문, 전자제품의 버튼 기호 등

III-1장

추상화

학습 목표

- 실생활 문제의 목표 상태를 파악하고 목표 상태 도달을 위한 문제 분석과 분해를 할 수 있다.
- 문제 상황의 패턴을 분석하고 핵심 요소를 찾아 다양한 방식으로 추상화 할 수 있다.

다양한 분야에서의 추상화 사례

일반적으로 추상화(abstraction)란 특정한 목적에 관련된 필수적인 정보만을 추출하고 기타 정보는 생략하거나 삭제하여 단순화시키는 것을 말한다. 컴퓨터에서 추상화 역시 이와 같은 의미로 복잡한 자료나 시스템 등으로부터 핵심 개념이나 기능을 간추려 표현하는 것을 말한다.

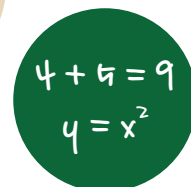
나무를 추상화하기 위해서 나무의 색채와 나뭇잎 형상을 배제하고 곡선으로 표현하다가 수직·수평선만을 이용하여 단순하게 표현할 수 있다.

사과의 개수를 4라는 숫자로 나타내고 여러 사람이 탄 사과를 합하여 '4 + 5 = 9'로 표현하는 것 역시 수와 연산자를 이용하여 추상화 한 것이다. 또한 음의 길이와 높낮이를 음표와 악보로 추상화할 수 있다.

사람들은 의사소통을 위해 상형문자에서 시작하여 오늘날과 같이 다양한 글자를 만들어 사용하며 자신의 뜻을 단순한 문자로 추상화하였는데 이는 언어의 추상화라고 볼 수 있다.



▲ 미술에서의 추상화



▲ 수학에서의 추상화



▲ 음악에서의 추상화



▲ 언어의 추상화

다음 만화를 보며 그림 암호를 풀어 빈칸에 적어 보자.

탐구 자료 부록 3(180p)



생·각·해·보·기

복잡한 의미를 간단한 그림이나 기호로 나타낸 사례를 찾아보자.

- 교과서 77쪽 동기 유발 ‘셜록 홈즈의 춤추는 인형 암호’ 만화를 읽어 보고 의미 전달을 위한 다양한 방법을 생각해 본다.

만화에 나오는 간단한 암호를 풀어 보고 의미 전달을 위해 간단한 그림으로 표현할 수 있음을 경험하게 한다.

▶ 암호 빈칸 답

- Never

춤추는 인형 그림은 인형의 자세 하나가 하나의 알파벳을 상징하고 이것은 미국 시카고의 악명 높은 범죄 집단에서 쓰이는 암호였다. 에이브 슬래니는 수배 중인 범죄자였고 의뢰인인 힐튼의 아내 엘시는 에이브 슬래니의 옛 연인이자 그 범죄 집단의 일원이었다. 그녀는 과거를 청산하기 위해 영국으로 넘어와 힐튼과 결혼했으나 자신을 데려가기 위해 찾아온 에이브 슬래니의 춤추는 인형 메시지를 보고 겁에 질려 있었던 것이다.

에이브 슬래니는 결국 집까지 들이닥쳤는데 엘시가 그를 설득해 떠날 것을 부탁하는 동안 힐튼이 두 사람을 발견하고, 총을 겨누었지만 결국 힐튼은 에이브 슬래니의 총을 맞고 사망하고 말았다. 엘시는 절망한 나머지 힐튼의 총으로 자살을 시도했지만 실패

해 중상을 입었다. 그러나 홈즈는 암호를 이미 해독했기 때문에 마치 엘시가 에이브 슬래니에게 보낸 것처럼 메시지를 역으로 남겨 그를 유인해 체포했다.



# 문제의 이해

## 핵심질문

문제를 해결하기 위해 현재 상태와 목표 상태를 파악하는 이유는 무엇일까?

- 문제를 해결을 위한 현재 상태와 목표 상태의 개념을 이해한다.
- 숫자 조각 퍼즐을 보며 현재 상태와 목표 상태를 이해한다.

## 참고자료

### 문제를 해결의 과정

문제의 현재 상태에서 목표 상태로 만들어 가는 과정이 문제 해결의 과정이다. 문제를 컴퓨터로 자동화하여 해결하기 위해서는 추상화의 과정이 필요하다. 추상화는 주어진 문제를 컴퓨터가 처리할 수 있는 언어로 변화시키기 위해 문제 이해, 문제 분해, 핵심 요소 추출의 과정을 거친다. 이렇게 문제를 추상화한 후에 일의 순서대로 절차를 만든다. 이렇게 일의 순서를 정한 것이 알고리즘이다. 이 알고리즘은 컴퓨터나 로봇이 처리할 수 있도록 프로그래밍 되어 목표 상태에 이르게 한다.



## 1-1 문제의 이해

현실 세계에 사는 우리는 직업, 교통, 환경, 안전, 의사소통, 금융 등과 관련된 다양한 문제 상황에 처할 수 있다. 컴퓨팅 시스템을 이용하여 이러한 현실 세계의 문제를 해결하기 위해서는 문제 상황을 어떻게 이해하고 분석해야 할까?

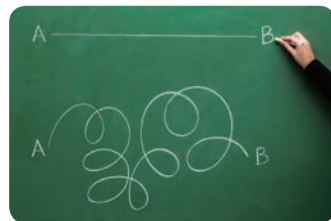
### 1 문제 상황 파악

문제를 해결해야 하는 일이나 과제를 말한다. 이러한 문제는 주어진 '현재 상태'와 원하는 '목표 상태' 사이에 차이가 있기 때문에 발생한다. 이와 같이 '현재 상태'를 '목표 상태'로 도달시키는 과정을 문제 해결 과정이라고 한다. 따라서 문제 해결을 위해서는 '현재 상태'와 '목표 상태'를 잘 파악해야 한다. 다음 퍼즐 문제를 생각해 보자. 왼쪽 퍼즐과 오른쪽 퍼즐의 차이는 무엇일까?

현재 상태	목표 상태
5	1
4	2
6	8
1	4
7	7
3	6
2	5

▲ 퍼즐 문제

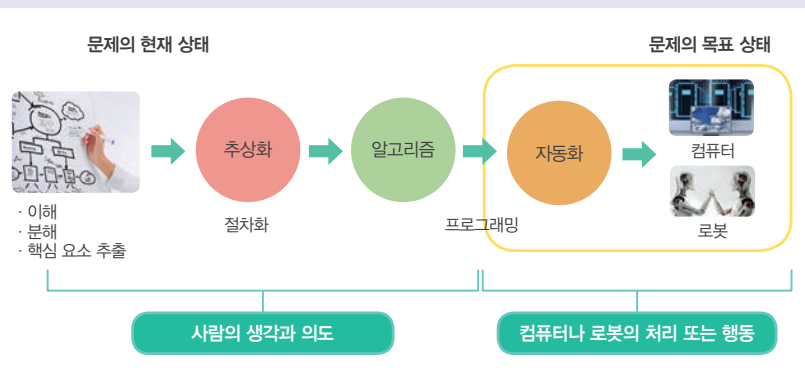
이 퍼즐 문제를 해결하기 위해 무작정 조각을 옮기기보다는 현재 상태와 목표 상태의 차이를 분석하는 것이 필요하다. 문제를 어떻게 파악하는지에 따라서 문제 해결 과정이 다르게 나올 수 있기 때문이다.



▲ 문제 해결 과정의 다양성

78

Ⅲ부 문제 해결과 프로그래밍

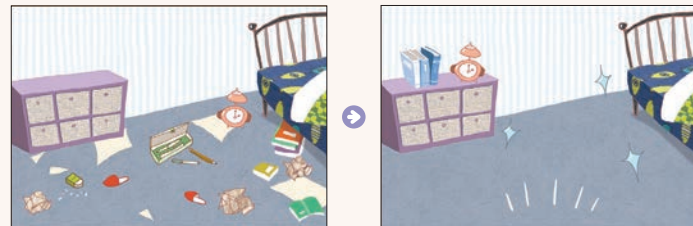


### 문제 상황 파악하기

#### 1 은아의 방 청소

다음 상황을 읽고 질문에 답해 보자.

은아는 자기 방에 누워 있다. 방으로 들어온 엄마는 지저분한 방을 보고 은아에게 청소를 하지 않으면 일요일 친구들과의 외출을 금하겠다고 하신다. 하지만 막상 지저분한 방을 치우려고 하니 은아는 고민이 된다.

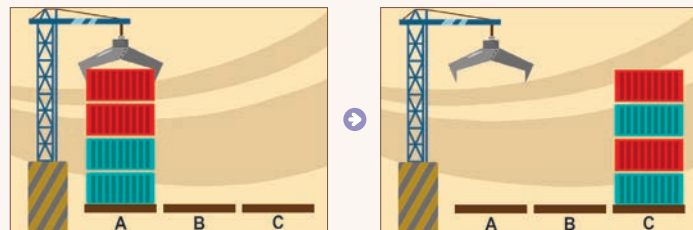


• 현재 은아의 방 상태는 어떠한가?

• 엄마가 원하는 상황은 어떤 것일까?

#### 2 컨테이너의 블록 옮기기

아래와 같이 왼쪽 그림의 컨테이너를 오른쪽 그림처럼 바꾸고자 한다. 어떻게 옮기면 좋을지 방법을 알아보자. 단, 한 번에 컨테이너 하나씩만 움직일 수 있다.



• 현재 상태

• 목표 상태

• 현재 상태와 목표 상태의 차이점

Ⅲ-1장 추상화

79

• 함께 해 보기의 '은아의 방' 상황에 대한 글과 그림을 살펴보고 현재 상태와 목표 상태를 알아본다.

• 함께 해 보기의 '컨테이너 옮기기' 문제의 현재 상태와 목표 상태를 파악하고 문제 해결 과정을 생각해 본다.

### 함께 해 보기 예시 답안

#### 1 은아의 방

- 현재 상황: 부스러기와 먼지가 떨어져 더러워진 방
- 원하는 상황: 바닥을 깨끗하게 치워야 함

#### 2 컨테이너 옮기기

- 현재 상태 파악: A라인에 파란색 컨테이너 2개와 빨간색 컨테이너 2개가 차례로 쌓여 있다.
- 목표 상태 파악: C라인에 파란색, 빨간색, 파란색, 빨간색 컨테이너 순으로 쌓여 있다.
- 현재 상태와 목표 상태의 차이점: 컨테이너들의 순서가 바뀌어 있다.

### • '컨테이너 옮기기'의 목표 상태로 가기 위한 방법 예시

(A↑ B↓: A에 있는 맨 위의 것을 B에 내려놓는다.)

A↑B↓ - A↑B↓ - A↑C↓ - B↑C↓ - A↑C↓ - B↑C↓

컨테이너가 많아질 경우 목표 상태로 가기 위한 경우의 수도 많아진다. 또한 같은 동작을 반복하게 되어 같은 해결 방법인데 더 오래 걸릴 수도 있다. 이와 같이 다양한 경우를 경험하고 이야기를 나누게 한다. 따라서 문제를 해결하는 복잡한 과정을 단순화할 수 있는 규칙을 찾아 정리할 필요가 있음을 알게 한다.

문제를 정확하게 이해하고 분석하는 것은 왜 중요할까?  
문제를 해결하기 위한 문제 분해 과정은 왜 필요할까?

- 은아네 집의 거실 상황을 파악하는 활동을 통해 문제 해결 과정에서 문제를 정확하게 이해하고 분석하는 것을 경험한다.
- 문제 해결을 위한 처음 전략으로 분해의 과정을 이해한다.

### 문제 분석 예시 답안

- 식탁과 의자같은 장애물이 있다.
- 청소기가 빨아들이기 어려운 옷 같은 물건들이 있다.
- 구석에 먼지가 많이 몰려 있다.

0과 1로 단순하게 처리하는 컴퓨터의 특성을 염두에 두고 문제를 데이터와 절차의 측면에서 작은 단위로 분해해야 프로그래밍할 때 용이하다는 것을 언급한다.

### 분해의 방법

- ▶ 데이터의 분해: 청소할 구역, 청소할 구성원 등의 객체나 변수로 지정할 수 있는 요소들을 찾아 분해할 수 있다.
- ▶ 절차(기능)의 분해: 역할을 분담하여 청소하기, 청소 구역의 순서를 정하여 청소하기 등과 같이 일의 순서나 기능별로 분해할 수 있다.

### 데이터 추상화

데이터의 추상화는 프로그램 작성자에게 프로그램에서 사용되는 데이터의 구조와 조작을 효과적으로 사용할 수 있도록 잘 정리된 수단을 제공하는 것이다. 현실 세계의 사물을 개념화, 단순화하고

## 2 문제 분석과 분해

### 1 문제 탐색 및 조사하기

문제 해결 과정에서 문제를 정확하게 이해하고 분석하는 것은 매우 중요하다. 이를 위해서는 세밀한 관찰이 필요하며, 관찰을 통해 주어진 문제에서 알아야 할 사실이나 자료가 무엇인지 발견할 수 있다.

#### 로봇 청소기의 문제 분석

은아는 거실을 청소하기 위해 로봇 청소기를 사용하기로 하였다. 한참 후 거실로 돌아와 보니 청소가 제대로 되어 있지 않았다. 은아는 로봇 청소기에 어떤 문제가 있는지 곰곰이 생각해 보았다.

- 거실 그림을 보면서 로봇 청소기가 작동하는 데 어떤 문제가 발생하였는지 추측해 보자.



▲ 은아네 거실

### 2 문제 분해하기

문제를 이해하고 나면 해결을 위해 문제의 분해 과정을 거친다. 문제의 분해란 주어진 문제를 간단한 작은 문제로 나누는 과정이다. 이렇게 작게 나누는 문제를 하나씩 해결하여 전체 문제를 해결한다.

또한 문제 해결에 사용되는 자료나 절차를 작게 분해하여 컴퓨터가 처리 가능하도록 하는 과정이 필요하다.

예를 들어 스마트폰의 애플리케이션을 구현할 때 통화, 인터넷, 음악 재생기, 카메라 등의 여러 가지 기능을 나누어 생각하고 이를 각각의 개발자에게 분업시킬 수 있다. 또한 카메라의 세부 기능도 더 작게 나누어 여러 프로그래머가 나누어 작업하도록 할 수 있다.



▲ 문제 분해와 분업

데이터적인 측면과 기능적인 측면으로 분리 정의하여 데이터에 대한 조작을 효과적으로 수행할 수 있는 수단을 제공해 주는 작업이다.

출처: [네이버 지식백과] 데이터 추상화(data abstraction, - 抽象化, 『IT용어사전』, 한국정보통신기술협회

☑ 일상생활에서 문제를 분해하여 컴퓨터로 해결하면 좋은 사례를 친구들과 찾아본다.

☑ 학교, 교실, 집, 마트 등의 주변에서 생각해 본다.



### 문제의 분해와 해결하기

#### 1 청소 문제 분해

은아와 엄마, 아빠가 집을 효율적으로 청소하기 위해 다음의 기준에 따라 문제를 분해하여 해결해 보자.



가. 청소해야 하는 구역

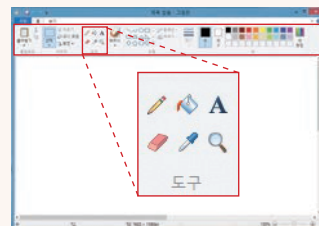
나. 청소할 수 있는 구성원

다. 어떤 청소 방식이 있는가?

- 걸레질을 한다.

#### 2 소프트웨어 모듈(기능) 분해

그림을 그리는 그림판 소프트웨어를 만들려고 할 때 어떤 기능으로 나눌 수 있을지 생각해 보자.



가. 그림판에서 사용되는 기능

나. 그림을 그리는 도구의 유형

#### 3 자동차 번호판 인식을 위한 분해

오른쪽 사진과 같이 자동차 번호판을 자동으로 인식하기 위해서는 자동차 본체와 번호판을 분해하여야 한다. 자동차 번호판을 컴퓨터가 인식하기 위해서는 어떤 식으로 분해해야 할까? 어떻게 나누어 인식하면 좋을지 선을 그어 보자.

34 가 8750



- 함께 해 보기 '청소 문제 분해' 활동을 통해 상황을 다양한 방식으로 분해해 본다.
- 함께 해 보기 '소프트웨어 모듈 분해' 활동을 통해 다양한 기능을 생각하여 분해해 본다.
- 함께 해 보기 '자동차 번호판 인식'을 위한 분해' 활동을 통해 패턴 인식을 위한 분해 활동을 해 본다.

다양한 접근의 문제 분해 과정을 경험하게 한다.

### 함께 해 보기 예시 답안

#### 1 청소 문제 분해

가. 방 3개, 거실 1개, 화장실 2개, 주방 1개.

나. 은아, 아빠, 엄마

다. 물건을 정리한다. 청소기를 돌린다. 설거지를 한다. 쓰레기를 묶어 버린다. 물청소를 한다. 등.

#### 2 소프트웨어 기능 분해

색 선택, 그리기 도구 선택, 도형 그리기, 선 그리기, 레이어 만들기 등

#### 3 자동차 번호판 인식을 위한 분해

34 가 8750

출 등에서 속도가 빠르고 정확한 처리가 가능하다는 장점이 있기 때문이다.

이진 영상에서 각 글자들의 위치가 추출되고 나면 글자 인식 단계로 들어간다. 이 단계는 글자의 표준 패턴과 이진 영상과를 정합하여 가장 유사한 글자를 고르는 원형 정합 방법, 글자의 구성 정보를 활용한 구문론적 방법, 신경망을 이용한 방법 등으로 계속 연구되고 있다.

12 1349

▲ 자동차 번호의 블록 이진화의 예

출처: 『대한민국 자동차 번호판에서 블록 이진화 기법을 이용한 문자 인식 기법』, 장인태, 이슬기, 우효정, 안재형 (2012), 컴퓨터정보통신연구, 20(1), pp.57-61

카메라 영상 입력

번호판 위치 추출

번호판 영역 이진화

이진 영상 전처리

글자 영역 추출

글자 인식





컴퓨터로 문제를 해결하기 위해 추상화 과정이 필요한 이유는 무엇일까?

- 다양한 사례를 통해 추상화의 개념을 이해하도록 한다.
- 프로그래밍에서 추상화를 하는 목적을 이해하도록 한다.
- 함께 해 보기를 통해 다양한 문제 상황에서 핵심 요소를 추출해 본다.

불필요한 요소와 필요한 요소를 구분하는 기준에 대해 토의하면서 핵심 요소의 개념을 이해하도록 지도한다.

### 함께 해 보기 예시 답안

- 1 공 튀기기  
공이 튕기는 높이, 공이 나아간 거리
- 2 정사각형 그리기  
앞으로 100 가기, 90도 돌기  
· 꼭 필요한 명령어: 펜 내리기, 100만큼 움직이기, 4번 반복하기, 90도 돌기  
· 불필요한 명령어: 소리 재생하기, 돌아! 말하기, 펜 색깔을 ■으로 정하기
- 3 스무고개 놀이  
정한 수, 상대방이 말한 수

- 객체(object, 오브젝트): 객체 지향 프로그래밍(OOP)이나 설계에서 데이터(실체)와 그 데이터에 관련되는 동작(절차, 방법, 기능)을 모두 포함한 개념

출처: 『IT용어 사전』, 한국정보통신기술협회

- 구현하고자 하는 상황에서 패턴을 알아보고 핵심요소를 찾는 것은 프로그래밍으로 구현할 때 필요한 객체, 변수, 리스트 등의 데이터 요소를 정하기 위해 매우 중요한 과정이다.



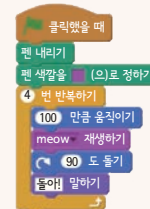
### 핵심 요소 찾기

- 1 그림과 같이 공이 튕기는 궤적의 패턴을 찾아 시뮬레이션을 만들고자 한다. 이 궤적의 패턴을 표현하는데 필요한 요소는 어떤 것인지 보기에서 찾아보자.



보기
공의 크기
공이 튕기는 높이
공의 구입 가격
공의 색깔
공이 나아간 거리
공이 멈춘 곳까지의 거리

- 2 로봇에게 정사각형을 그리게 하기 위해 다음과 같은 명령어를 작성하였다. 정사각형을 그리기 위해 필요한 명령어와 불필요한 명령어로 나누시오.



필요한 명령어	불필요한 명령어
100만큼 움직이기	소리 재생하기
90도 돌기	돌아! 말하기
4번 반복하기	펜 색깔을 <span style="background-color: black; color: white;">■</span> 으로 정하기

- 3 1~100 사이에서 하나의 수를 정하고 스무고개 놀이처럼 상대방이 그 수를 맞게 하려고 한다. 수를 정한 사람은 상대방이 말한 수와 비교하여 큰 수인지 작은 수인지만 알려줄 수 있다. 가장 적은 횟수로 그 수를 맞히기 위한 규칙을 찾을 때의 필요한 핵심 요소를 찾아보자.



### • 교육부(2015)의 컴퓨팅 사고력(computational thinking)

CT 용어	용어 정의
자료 수집	문제 해결에 필요한 자료 모으기
자료 분석	자료의 이해, 패턴 찾기, 결론 도출하기
구조화	문제를 그래프, 차트, 그림 등으로 시각화하기
분해	문제를 관리 가능한 수준의 작은 문제로 나누기
추상화	문제 해결을 위한 핵심 요소를 추출하고, 모델 만들기
알고리즘	문제를 해결하기 위한 일련의 단계를 알고리즘으로 표현하기(절차적 표현)

## 2 추상화

### 1 추상화의 개념과 이해

발견한 패턴을 바탕으로 필요한 것에 집중할 수 있도록 중요한 것은 남기고 무시할 것은 걸러 내어 단순하게 표현하는 과정을 추상화라고 한다. 문제 상황이나 전달하고자 하는 내용을 정리하지 않고 상대방에게 모든 세부 내용을 설명하다 보면 이해하기 쉽지 않고 핵심 내용이 제대로 전달되지 않을 수 있다. 반면 문제의 핵심을 간단한 도식, 표, 공식 등으로 정리하여 전달하면 쉽게 이해시킬 수 있다.



컴퓨터에서의 추상화(abstraction)는 주어진 문제에서 필요 없는 내용은 빼고 핵심적인 개념 또는 기능을 간추려 그 특징과 패턴을 알기 쉽게 표현하는 것이다.

추상화는 문제를 알고리즘으로 표현하고 프로그래밍하기 위해서도 반드시 필요한 과정이다. 추상화는 생각의 단위를 법칙이나 공식으로 표현할 수도 있고 동시에 그것이 무엇을 의미하는지 알게 해 준다. 추상화의 목적은 다음과 같다.

- 문제가 명확해지고 규칙적이 되어 이해하기가 쉽다.
- 프로그램의 개발과 유지 보수가 쉽다.
- 프로그램의 확장성이 쉽다.

추상화는 작은 수준의 의미를 함수로 표현하는 것이므로 작은 수준으로 문제를 분해하고 패턴을 인식하는 과정이 필요한 것이다. 즉 추상화는 단순하게 만드는 것이 핵심이며, 하나의 함수는 하나의 일을 하도록 해야 한다.

함수  
어떤 재료를 넣어 새로운 값을 만드는 틀



## • 일상생활에서의 추상화 예

### 1 반복되는 일을 추상화한 시간표

수요일	월	화	수	목	금
09:00 1 09:50	역사	생물	국어	수학	영어
10:00 2 10:50	H123	G221	E401	G234	E225
11:00 3 11:50					
12:00 4 12:50					
13:00 5 13:50	수학	영어		역사	국어
14:00 6 14:50	G106	E225		H125	E403
15:00 7 15:50					
16:00 8 16:50	국어	수학	생물		
17:00 9 17:50	E401	G106	G222		

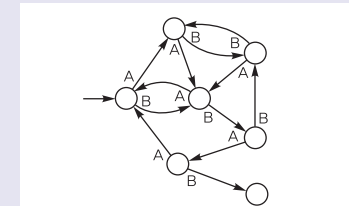
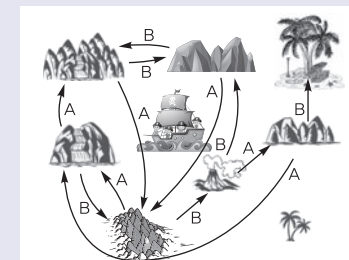
▲ 편리한 시간표 앱 화면(<https://play.google.com/store/apps/details?id=com.newbitmobile.handytimetable&hl=ko>)

### 2 지하철 노선도



▲ 네이버 지하철 노선도(<http://map.naver.com/?menu=subway&region=1000>)

### 3 오토마타 향해 지도를 보기 쉽게 도식화함



▲ CS Unplugged, Tim. Bell, 2006



# 핵심 질문

프로그램을 위한 추상화의 방법에는 어떤 것들이 있을까?

- 다양한 사례를 통해 추상화의 방법을 경험하도록 한다.

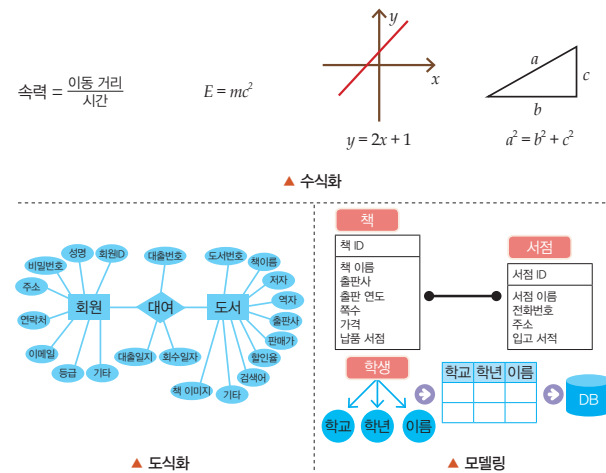
만화의 내용과 같이 속력을 계기판에 표시 되도록 구현하기 위해 속도, 이동 거리, 시간의 핵심 요소 간의 관계를 식으로 간단하게 추상화하여 나타낼 필요가 있음을 이해시키도록 한다.

- 추상화 표현 방법**
  - 명명화
  - 수식화
  - 모델링
  - 도식화
  - 기호화

- 모델링(modeling)**
  - 현실 세계에 존재하는 자료(데이터)를 컴퓨터 세계의 데이터베이스로 옮기는 변환 과정

## 2 추상화의 표현

추상화의 표현 방법에는 다음과 같은 다양한 방법이 있다.



86

Ⅲ부 문제 해결과 프로그래밍

참고자료

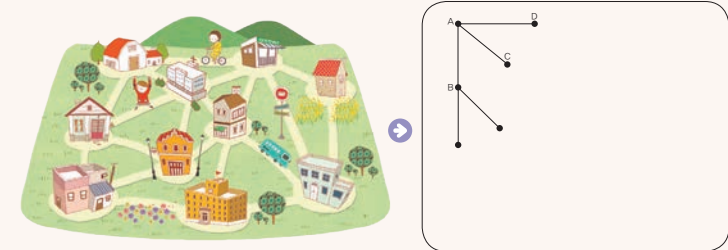
## 추상화의 표현 방법

표현 방법	개념
수식화	<ul style="list-style-type: none"> <li>패턴 간의 관계를 수식으로 표현</li> <li>패턴에서 보이는 조건을 공식(수학적 표현)으로 표현</li> </ul>
도식화	<ul style="list-style-type: none"> <li>문제 상황을 단순한 기호로 표시하거나 문제의 상호 관계를 단순한 도형과 선, 화살표 등으로 나타냄</li> </ul>
모델링	<ul style="list-style-type: none"> <li>반복되는 일(패턴)의 추상화</li> <li>문제 해결을 위한 문제에 대한 기본적인 개념 틀을 만드는 것</li> <li>모델을 만들어 두고 하위 요소를 변경하면 다양한 데이터가 됨</li> <li>문제에 대한 모델을 하나 만들면 문제를 볼 수 있는 절차와 방법을 설계하여 다양한 문제를 해결</li> </ul>



## 추상화하기

1 아래 그림은 마을 지도이다. 건물은 알파벳과 점으로, 도로는 선으로 표기하자. 마을 지도를 간단하게 나타내기 위해 건물과 도로를 선과 점의 도식으로 그려 보자.



2 교통 수단의 속력을 측정하기 위해 이동 거리, 시간과 속력의 관계를 관찰하고 그 패턴을 찾아 수식으로 나타내 보자.

	이동 거리(km)	시간(h)	속력(km/h)
자전거	55	5	11
자동차	100	2	50
기차	560	4	140

3 다음은 다양한 몬스터를 그리기 위한 지시문이다. 다음 빈칸에 원하는 숫자를 넣고 짝에게 지시문에 맞게 그리도록 해 보자.

몬스터 그리기 지시문	지시문을 따라 그린 몬스터
____ 모양의 얼굴    ____ 개를 그리시오. ____ 모양의 입    ____ 개를 그리시오. ____ 모양의 코    ____ 개를 그리시오. ____ 모양의 눈    ____ 개를 그리시오. ____ 모양의 귀    ____ 개를 그리시오.	

위 지시문을 아래와 같이 표로 나타내고자 할 때 행과 열의 머리글로 알맞은 요소를 적어 보자.

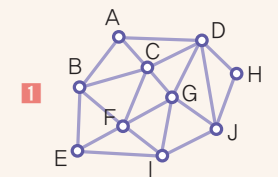
	세모	네모	동그라미	별	물결
	1	2	4	2	3

Ⅲ-1장 추상화

87

- 함께 해 보기를 통해 추상화의 다양한 방법을 연습한다.
- 핵심 요소를 이용하여 규칙성을 다양한 방법으로 표현해 보도록 한다.
- 컴퓨터로 처리하기 위해 추상화 과정이 왜 필요한지 0과 1로 처리하는 컴퓨터의 특징과 관련지어 설명한다.

## 함께 해 보기 예시 답안



2 속력 =  $\frac{\text{이동거리}}{\text{시간}}$

	얼굴	입	코	눈	귀
모양					
개수					

- 몬스터 그리기를 통한 모델링**  
정해진 틀 안에서 숫자만 바꾸면 다양한 몬스터가 나온다. 즉, 다양한 몬스터가 만들어지는 프로그램을 만든 것이다. 이와 같이 추상화된 요소를 가지고 일정한 패턴을 나타내는 모델을 만들면 여러 결과를 가질 수 있는 일정한 양식을 만들 수 있는데 이것이 모델링이다.

## 추상화 설명의 다른 예

토끼를 본 적이 없는 사람에게 토끼를 설명하려고 할 때, 다음의 왼쪽 글과 같이 말할 수 있을 것이다. 설명을 들은 사람들은 토끼를 오른 쪽과 같이 정의하게 될 것이다.

“토끼는 다리가 네 개이다. 뒷다리는 앞다리보다 길고 튼튼해서 오르막을 잘 올라갈 수 있다. 그리고 귀가 두 개 있는데 그 귀의 길이가 매우 길다. 그리고 그 귀를 이용해서 멀리 있는 소리를 잘 듣고 맹수의 위협에서 벗어날 수 있다. 토끼는 산토끼와 굴토끼가 있다.”

- 1 다리가 네 개
- 2 길고 튼튼한 뒷다리
- 3 긴 두 개의 귀
- 4 멀리 있는 소리도 잘 들을 수 있음.
- 5 산토끼와 굴토끼가 있음.

위와 같이 토끼가 무엇인지에 대한 간단한 정의가 내려지는데 이것도 추상화의 일종이다. 이런 추상화를 통해 실세계의 사물을 데이터적인 측면과 기능적인 측면으로 정의할 수 있다.

## 실습 활동 & 토론하기

• 실습 활동을 통해 학습 내용을 정리해 본다.

**지도상의 유의점** [1-1 문제 이해] 단원과 [1-2 핵심 요소 추출] 단원의 핵심 내용을 모두 연결해서 이해하도록 한다.

### 실습 활동 답안

- 1 현재 상태: 왼쪽 아래에서 대기  
목표 상태: 사각형 모양으로 돌며 움직이기
- 2 먼저 흡입하기, 움직이기, 직진하기, 돌기, 장애물 감지하기 등
- 3 벽에 닿을 때까지 직진하기, 90도 돌기
- 4 직진하기, 돌기
- 5 직진하고 벽에 닿으면 90도 돌기, 4번

### 토론하기

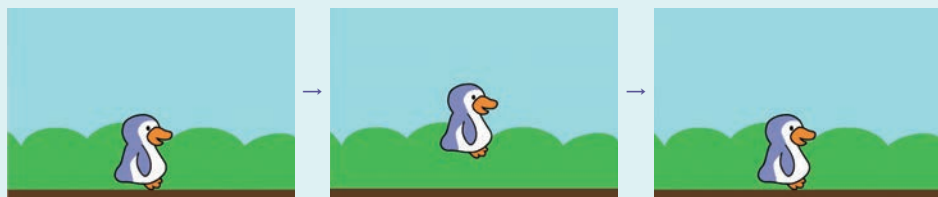
- 1 공통점: 전달하고자 하는 의미를 단순한 이미지나 기호로 표시한다.
- 2 차이점: 컴퓨터 과학에서는 프로그래밍 언어로 표현할 수 있도록 데이터와 기능의 핵심 요소를 찾아 추상화한다.

### 참고자료

#### 추상화 설명의 다른 예

프로그램 세부 모듈에서의 기능 추상화의 예

- ▶ ‘점프하기’라는 명령은 객체를 위로 올렸다가 내려가게 한다는 의미를 포함한다. 또한 여기서 객체를 올리거나 내린다는 것은 점프하기의 핵심 요소인 객체의 ‘Y좌표 값’을 (+) 값과 (-) 값으로 변화시키는 것으로 추상화할 수 있다(<https://scratch.mit.edu/projects/119947695>).

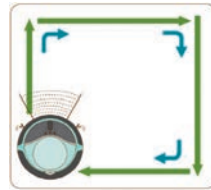


스페이스 키를 눌렀을 때  
Y 좌표를 50 만큼 바꾸기  
0.2 초 기다리기  
Y 좌표를 -50 만큼 바꾸기

## 실습 활동

오른쪽 그림과 같이 로봇 청소기가 사각형 모양으로 이동하도록 하고자 한다. 다음 질문에 알맞은 답을 <보기>의 단어를 이용하여 쓰시오.

**보기**  
직진하기, 돌기, 왼쪽 아래, 사각형 모양



1 문제 상황의 현재 상태와 목표 상태를 설명하시오.

현재 상태	에서 대기
목표 상태	모양으로 돌며 움직이기

2 문제 분해: 그림에서 나타나는 로봇 청소기의 세부 동작은 , . 장애물 감지하기 등이다.

3 패턴 찾기: 로봇 청소기가 사각형을 그리며 방을 도는 과정에서 반복되는 패턴은 벽에 닿을 때까지 , 90도 이다.

4 핵심 요소 찾기: 로봇 청소기가 사각형을 그리며 움직이게 하는 기능에 대한 핵심 동작 요소는 와 이다.

5 추상화: 로봇 청소기는 ‘직진하다 벽에 닿으면 90도 ’의 패턴을 4번 반복한다.

### 자기평가

평가 내용	잘함	보통	노력 요함
1. 실생활 문제 상황을 이해하여 적게 분해할 수 있다.			
2. 문제 상황에서 핵심 요소를 파악할 수 있다.			
3. 목표 상태에 도달하기 위한 패턴을 추상화할 수 있다.			

### 토론하기

미술, 수학, 음악, 언어에서의 추상화와 컴퓨터 과학에서 추상화의 공통점과 차이점에 대해 토의해 보자.

공통점	차이점
1	2

88

Ⅲ부 문제 해결과 프로그래밍

### 배운 내용 정리하기



### 평가하기

※ 다음 문제 상황을 읽고 질문에 답해 봅시다. (1~3)

우리 가족은 주말마다 농장에 가서 감자와 고구마 등을 심어서 가꾸고 있다. 주말에만 농장에 갈 수 있어 주중에는 물을 주기 어렵다. 주중에도 땅이 마르거나 날이 더우면 자동으로 물을 주는 스프링클러가 필요하다.

1 위 문제 상황에 대한 목표 상태는 무엇인가?

2 스프링클러가 자동으로 작동하기 위한 핵심 요소는 무엇인가?

3 스프링클러, 습도 센서, 온도 센서 등의 관계를 추상화하여 도식화해 보시오.

Ⅲ-1장 추상화

89

## 배운 내용 정리 & 평가하기

### 평가하기 예시 답안

- 1 땅의 습도나 기온에 따라 스프링클러가 자동으로 작동하도록 하는 것
- 2 땅의 습도, 기온
- 3 습도 센서 값 < 20 ? → 작동  
온도 센서 값 > 35 ? → 작동

• 피지컬 컴퓨팅을 활용하여 땅의 수분을 측정할 수 있음을 이해하여 Ⅳ부의 피지컬 컴퓨팅과 연계하여 지도할 수 있다.

• 실제로 다양한 센서를 이용하여 화훼 및 농업 분야에서 자동 스프링클러와 같이 도움을 줄 수 있는 사례들을 찾아보자.

### 참고자료

#### 스프링클러 문제의 피지컬 컴퓨팅의 적용

스프링클러가 땅의 습도에 따라 작동 여부를 결정하기 위한 실험을 위해 피지컬 컴퓨팅을 사용할 수 있다. Ⅳ부 피지컬 컴퓨팅 지도서 232쪽을 참고하여 센서 보드의 저항선을 이용하여 땅의 습도를 수치로 측정하고 스프링클러의 작동 여부를 위한 기준 값을 설정할 수 있다.

## 생각해 보기

- 은아와 지원은 게임 방법을 명확히 하기 위해 어떻게 하였을까?
- 은아는 지원이 여러 번 반복하여 적은 규칙을 왜 한 번만 적자고 하였을까?
- 일상생활에서 일련의 순서와 규칙을 정해서 해야 하는 일들의 예를 이야기 해 본다.



- 학습 목표**
- 사람과 컴퓨터의 문제 해결 방법의 차이를 설명할 수 있다.
  - 알고리즘의 개념을 이해하고 중요성을 인식할 수 있다.
  - 알고리즘을 다양한 방법으로 표현하고 설계할 수 있다.

## 주사위 게임 규칙 만들기

은아와 지원은 주사위 게임을 직접 만들어서 놀기로 하고 게임 방식을 상의하여 다음과 같이 규칙을 정하였다.

	TURN 1	TURN 2	TURN 3	TURN 4	TURN 5	TURN 6	TOTAL
Player 1	_____	_____	_____	_____	_____	_____	_____
Player 2	_____	_____	_____	_____	_____	_____	_____

1. 가위바위보를 해서 누가 먼저 주사위를 굴릴지 결정한다.
2. 모든 사람의 점수는 0으로 시작한다.
3. 한 사람이 주사위를 굴린다.
4. 주사위에 나온 숫자를 이전 숫자에 더한다.
5. 다음 사람이 주사위를 굴린다.
6. 규칙 4~5 과정을 여섯 번 반복한다.
7. 최종 숫자를 확인한다.
8. 숫자가 큰 사람으로 승자를 결정한다.

게임 규칙을 정하는 과정에서 처음에 지원이 규칙 4~5번 과정을 총 6번 반복하여 적어 놓았다. 은아가 반복되는 규칙인 4~5번은 한 번만 적고 규칙 6번을 은아가 추가하자 규칙의 개수가 줄고 이해하기 쉬워졌다.

이렇게 문제를 해결하기 위한 일련의 절차 및 순서를 알고리즘이라고 한다. 이러한 순서와 과정이 명확히 제시되지 않는다면 주어진 일을 제대로 수행할 수 없을 것이다. 주어진 일을 처리하는 방법은 다양하지만 일을 신속하고 정확하게 처리하기 위해서는 알고리즘을 효율적으로 만들 필요가 있다.

다음 만화를 보며 정보를 효율적으로 처리하는 데 알고리즘이 왜 필요한지 알아보자.



## 생각해 보기

위의 만화를 보고 알고리즘이 적용된 실생활의 예를 더 찾아보자.

**알파리조미:** 페르시아의 수학자로 '대수학의 아버지'로 불리며 알고리즘이란 말은 그의 이름에서 나왔다.

- 수십여 건의 문서 검색, 여러 명에게 메일을 동시에 보내는 일, 온라인 쇼핑에서의 전자상거래, 수백 개의 사진 파일 압축, 검색어를 잘못 입력해도 오타를 수정해 주는 작업 등 이러한 편리한 기능들을 사용할 수 있게 된 데는 아이디어를 정확한 절차와 방법으로 구체화시킨 효율적인 알고리즘 설계의 영향이 컸음을 설명하며, 위의 예시를 제외한 실생활에서 적용되고 있는 효율적인 알고리즘 사례에 관해 더 이야기를 나누어 본다.
- 동기 유발 자료를 통해 위대한 아이디어를 구체화시킨 알고리즘 적용 사례에 대해 이야기해 보고, 이 밖에 다른 사례가 있는지 발표해 본다.

## • 다양한 알고리즘의 예

알고리즘 종류	내용
검색 알고리즘	대량의 데이터에서 원하는 정보를 찾아낼 때 빠르고 효과적인 방법이다.
압축 알고리즘	문서, 소리, 이미지, 영상 등의 데이터 크기를 줄이는 목적으로 가공하는 방법이다.
정렬 알고리즘	데이터를 일정한 규칙에 따라 재배열하는 방법이다.
암호화 알고리즘	원문과 암호화 키를 입력하여 암호화된 데이터를 출력으로 생성하는 알고리즘이다.
얼굴 인식 알고리즘	얼굴의 주요 특징을 추출해 비교 분석하는 방법이다.
최단 거리 알고리즘	두 지점 사이의 가장 짧은 거리를 계산하는 방법이다.



# 알고리즘의 이해

2-1

## 핵·심·질·문

알고리즘은 무엇이며 효율적인 알고리즘을 작성하기 위해서는 어떤 조건이 필요할까?

- 우리의 생활 주변에서 효율적으로 문제를 해결한 절차와 방법에 대해 다양한 경험을 공유하는 시간을 갖는다.
- 알고리즘의 개념을 정확하게 이해하고 효율적인 알고리즘을 작성하기 위해서는 어떤 조건이 필요한지 예를 들어 설명한다.

## 참/고/자/료

- 알고리즘이 무엇인지에 관해 설명할 때 우리 일상생활에서 쉽게 접할 수 있는 예제를 들어 설명하면 훨씬 이해도를 높일 수 있다. 예를 들어 학급별 테마형 체험 학습에 관한 계획을 수립한다고 할 때 이 문제를 해결하기 위한 방법과 절차는 다음과 같다.

담임 선생님과 함께  
여행 일자를 결정한다.

→ 학급 구성원의 의견을  
반영하여 최종적으로  
여행 장소를 결정한다.

→ 여행 일자에 숙박 시설의  
예약이 가능한 장소를 조  
사한다.

→ 예약 가능한 장소 중 체  
험 활동 시설이 가장 좋은  
곳을 조사한다.

- 임의의 숫자 찾기 게임에서 아무 카드나 무작정 뒤집어 볼 경우 첫 번째로 87을 찾을 수도 있고, 10번 만에 87을 찾을 수도 있다. 또한 가장 앞에 있는 숫자부터 카드를 뒤집어 볼 경우 8번 만에 87을 찾을 수 있다. 가장 가운데 있는 카드부터 뒤집어 볼 경우 2~3번 만에 87을 찾을 수 있다.



## 2-1 알고리즘의 이해

유명인의 프로필을 찾고자 할 때 검색 엔진은 어떠한 절차로 적절한 검색 결과를 보여 줄까? 스마트폰으로 지문을 인식하는 절차는 어떤 방법을 사용할까? 이러한 문제를 해결하는 효율적인 절차와 방법에 대해 알아보자.

### 1 알고리즘의 개념 및 조건

알고리즘이란 문제를 해결하기 위한 방법을 순서대로 나열한 절차이다. 일 상생활에서 해결해야 할 문제에 직면했을 때 해결 과정을 구체적으로 계획 한 후에 실행해야 한다. 이때 여러 가지 해결 방법과 단계별 순서를 생각해 보고, 그 중 가장 적합한 방법을 선택하여 실행해야 한다.

여러 개 나열된 숫자 카드 중에서 임의의 숫자를 어떻게 찾아낼까?

#### 임의의 숫자 찾기

제시된 문제 상황: 10장의 뒤집힌 카드에서 임의의 숫자 87을 찾는 방법에는 어 떤 것들이 있을까?

뒷면										
앞면	43	19	65	32	14	50	90	87	79	25

찾는 방법 적어 보기

- 1 아무 카드나 무작정 뒤집어 본다.
- 2 가장 앞에 있는 카드부터 뒤집어 본다.

3

4

좋은 알고리즘이 갖추어야 할 조건은 입력과 출력이 있어야 하며, 유한성, 수행 가능성 그리고 명확성이다.

▲ 알고리즘의 조건

92

Ⅲ부 문제 해결과 프로그래밍

### 2 알고리즘의 효율성

주어진 문제에 대해 여러 가지의 해결 방법이 있는 것처럼 알고리즘도 여 러 종류의 알고리즘이 있을 수 있으나 보다 효율적인 알고리즘을 고안하는 것이 중요하다. 똑같이 주어진 일이라도 어떠한 방식을 사용하느냐에 따라 비용과 효율성에서 많은 차이가 나기 때문이다.

정렬되어 있는 숫자 카드를 찾는 두 가지 방법을 비교해 보자.

#### 오름차순으로 정렬된 카드에서 임의의 숫자 찾기

제시된 문제 상황: 10장의 뒤집힌 카드에서 임의의 숫자 87을 찾는 방법에는 어 떤 것이 있을까?

14	19	25	32	43	50	65	79	87	90
----	----	----	----	----	----	----	----	----	----

▶ 10장의 카드가 오름차순으로 정렬되어 있을 때 숫자 87을 찾아내는 방법은 어 떤 것이 더 효율적일까?

#### 방법 1 순서대로 찾기

▶ 10장의 카드를 왼쪽부터 순서대로 읽어가며 탐색

1	2	3	4	5	6	7	8	9	
14	19	25	32	43	50	65	79	87	90

14 < 87, 19 < 87, 25 < 87, 32 < 87, 43 < 87, 50 < 87, 65 < 87, 79 < 87, 87 = 87

▶ 방법 1 순차 탐색(sequential search)

정렬되어 있는 자료를 순서대로 탐색하는 방법

▶ 방법 2 이진 탐색(binary search)

오름차순으로 정렬된 자료를 반으로 나누고, 나누어진 반을 다시 반으로 나누는 과정을 반복하여 원하는 결과를 찾는 탐색 방법

#### 방법 2 반으로 나누어 찾기

- ▶ 중간에 있는 카드의 숫자인 43이나 50과 비교
- ▶ 43은 87보다 작으므로 43보다 오른쪽에 있는 자료들을 탐색
- ▶ 오른쪽에 있는 값을 다시 반으로 나누어 65나 79를 87과 비교해 가면서 87을 탐색

14	19	25	32	43	50	65	79	87	90
				43 < 87					
					50	65	79	87	90
						65 < 87			
							79	87	90
								79 < 87	
									87 = 87
								87	90

Think

▶ 방법 1 과 방법 2 는 87을 찾기 위해 각각 몇 번의 비교 횟수가 필요한가?

● 오름차순 정렬  
알파벳 또는 숫자 자료를 작은 것부터 큰 것의 순으로 나열하는 방법이다. 반대의 경우는 내림차순 정렬이라고 한다.

Ⅲ  
2장  
알고리즘

## 핵·심·질·문

효율적인 알고리즘을 고안하는 것은 왜 중요할까?

- 알고리즘 설계에 따라 비용과 효율성에 많은 차이가 생긴다는 것을 이해하도록 돕는다.
- 정렬되어 있는 숫자 카드를 찾는 방법 중 순서대로 찾는 방법과 반으로 나누어 찾는 방법을 비교하여 탐색 횟수에 얼마나 차이가 나는지 확인한다.
- 알고리즘의 비용이란 시간 복잡도와 공간 복잡도를 의미한다.

## Think 답안

방법 1 9번

방법 2 4번

## • 순차 탐색

정렬되어 있는 자료를 순서대로 탐색하는 방법이다. 만약 10장의 카드가 오름차순으로 정렬되어 있을 때 순차 탐색으로 숫자 87을 찾아내는 방법은 14, 19, 25, 32, 43, 50, 65, 79를 순서대로 읽어가며 9번째 탐색을 해야 87을 찾을 수 있다.

## • 이진 탐색

오름차순으로 정렬된 자료를 반으로 나누고, 나누어진 반을 다시 반으로 나누는 과정을 반복하여 원하는 결과를 찾는 탐색 방법이다. 만약 10장의 카드가 오름차순으로 정렬되어 있을 때 이진 탐색으로 숫자 87을 찾아내는 방법은 중간에 있는 카드 43과 87을 비교하여 87은 43보다 크므로 43보다 오른쪽에 있는 자료를 탐색하고, 오른쪽에 있는 값을 다시 반으로 나누어 65와 87을 비교하여 오른쪽에 있는 자료를 탐색하고, 79와 87을 비교하여 탐색 횟수 4번 만에 87을 찾을 수 있다.

## • 순차 탐색과 이진 탐색의 특징

### 순차 탐색의 특징

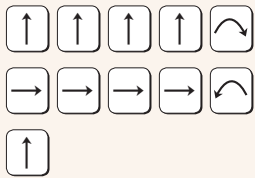
- 탐색 방법이 단순함.
- 자료가 정렬되어 있지 않아도 사용 가능.
- 자료의 양이 많거나 탐색을 빨리 해야 하는 경우 적당하지 않음.

### 이진 탐색의 특징

- 순차 탐색에 비해 복잡.
- 이진 탐색을 실시하기 전에 탐색 속도를 높이기 위해 자료를 미리 정렬해야 함.



이동할 순서



탐구 과제

- 화살표 개수  
표기 횟수가 적은 것 등
- ↑ ×4 ↻ → ×4 ↻ ↑

- 친구들이 해결한 알고리즘과 비교하여 이외에도 다양한 방법이 있음을 이해하게 한다.
- 함께 해 보기에서는 똑같은 동작이 반복될 경우 순차적으로 여러 번 표현하였으나 탐구 과제에서는  $\times 2$  2회 반복,  $\times 4$  4회 반복이라는 반복 명령어를 추가하여 알고리즘을 훨씬 간단하게 표현할 수 있음을 알게 한다.

참/고/자/료

· 추가 선택 활동 ·

1 아래와 같이 화살표 명령어를 이용하여 알고리즘을 나타내 보자(별표가 시작점).

→	←	↑	↓	↻
오른쪽으로 한 칸 이동	왼쪽으로 한 칸 이동	위쪽으로 한 칸 이동	아래쪽으로 한 칸 이동	해당 칸을 한 칸 색칠하기

2 아래 그림을 그리기 위한 화살표 알고리즘을 적어 보자.

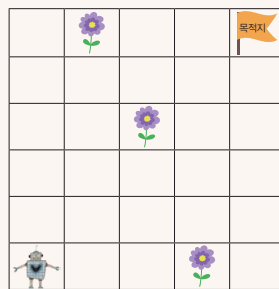
★										
	■									
		■								
			■							
				■						
					■					
						■				
							■			
								■		
									■	

숫자 카드 중 임의의 수를 찾아내는 사례에서 보듯이 어떤 알고리즘을 사용하느냐에 따라 탐색 횟수가 9번이 될 수도 있고 4번이 될 수도 있다. 숫자 카드가 많아질수록 탐색(비교) 횟수에 더 많은 차이가 난다. 수많은 자료를 처리하는 컴퓨터의 경우 대량의 자료를 반복 처리하는 일을 하므로 알고리즘을 효율적으로 작성해야만 합리적으로 문제를 해결할 수 있게 된다. 예를 들어 1,000만 건의 자료를 처리하는 데 있어 한 건당 0.1초씩의 시간 손해가 발생했다고 하면 무려 11일의 시간 손해가 발생하게 된다.

따라서 효율적인 알고리즘이 컴퓨터의 성능과 소프트웨어의 처리 비용에 중요한 영향을 미친다.

로봇을 목적지까지 이동시키는 알고리즘

로봇을 목적지까지 이동시키려 한다. 로봇은 한 번에 한 칸씩 이동 가능하며 은 장애물이다. 명령어를 이용하여 목적지까지 이동하도록 순서대로 알고리즘을 만들어 보자. ( 장애물은 피해 가야 한다.)



명령어

- ↑ 위로 1칸 이동
- ↓ 아래로 1칸 이동
- ← 왼쪽으로 1칸 이동
- 오른쪽으로 1칸 이동
- ↻ 오른쪽으로 90도 회전
- ↻ 왼쪽으로 90도 회전

이동한 순서의 명령어 또는 이동 기호

- 탐구 과제: 1 친구들이 해결한 알고리즘 중 누구의 알고리즘이 더 나은지 비교해 보자.  
2 누구의 알고리즘이 좋은지 비교해 보는 기준은 무엇인가?  
3 다음과 같은 명령어가 추가된다면 로봇의 이동 알고리즘은 어떻게 될까?

$\times 2$  2회 반복  $\times 4$  4회 반복

3 아래 화살표 알고리즘을 이용하여 사각형 안에 색을 칠해 그림을 완성해 보자.



→	↻	→	↓	↻	←	↓	↻	→	↓
1단계	2	3	4	5	6	7	8	9	10
↻									
11									

## 2-2 알고리즘의 표현

음악을 연주하기 위해 악보로 표현하고 수학 문제를 해결하기 위해서 공식으로 표현한다. 컴퓨터로 문제를 해결하기 위한 알고리즘은 어떻게 표현하는지 살펴보자.

### 1 알고리즘 표현 방법

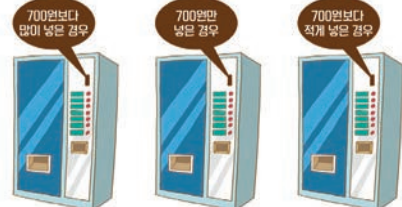
알고리즘을 표현하는 방법은 다양하나 일반적으로 자연어, 의사 코드, 순서도 등의 방법을 사용한다.

#### 1 자연어

자연어 표현 방법은 우리가 일상생활에서 사용하는 언어로 알고리즘을 표현하는 방식이다. 자판기의 작동 알고리즘과 짝수·홀수를 판단하는 알고리즘을 자연어로 표현한 예를 살펴보자.

##### 자판기의 동작 원리

700원짜리 음료수 버튼이 눌러지면 동전이 투입되길 기다린다. 동전이 투입되었을 때 700원보다 많으면 음료수와 잔돈을 돌려준다. 700원과 같으면 음료수만 주고, 700원보다 적으면 음료수를 내보내지 않고 받은 돈을 돌려준다.



##### · 자연어 표현 방식

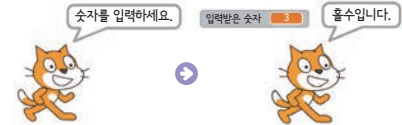
인간의 언어를 사용하기 때 문에 쉽게 이해할 수 있다는 장점이 있지만, 읽는 사람에 따라 모호한 해석이 이루어질 수 있으므로 최대한 명료하게 표현해야 한다.

☐ 사과의 크기를 대략 300g 초과하는 데, 300g~250g 사이는 중, 250g 미만은 소로 구분하자.

##### 짝수·홀수의 판별

임의의 숫자를 입력받아 그 값을 2로 나눈 후 나머지가 0이면 짝수, 아니면 홀수를 출력한다.

##### 문제 해결 순서



## 2-2

## 알고리즘의 표현

### 핵·심·질·문

알고리즘을 표현하는 방법에는 어떤 것이 있을까?

- 우리의 생활 주변에서 음악을 연주하기 위해 악보를 사용하고, 수학 문제를 해결하기 위해 공식을 표현하는 것처럼 문제를 해결하는 다양한 표현 방법 중에 다른 사례가 있는지 발표해 본다.
- 우리 생활에서 쉽게 접할 수 있는 자판기 작동 알고리즘, 짝수/홀수 판별 알고리즘의 예를 자연어로 표현하여 설명한다.
- 자연어로 표현하는 방법이 의사 코드, 순서도로 표현하는 방법과 비교할 때 장단점이 무엇인지 설명한다.

### 자연어 표현

#### 스프링클러의 작동 원리

스프링클러의 시작 버튼이 눌러져 있는지 확인한다. 시작 버튼이 눌러져 있다면 광도가 30 미만이면 먼저 저항이 80 이상이면 스프링클러를 작동시킨다. 광도가 30 이상이거나 저항이 80 미만이면 스프링클러를 작동시키지 않는다.



순서도는 어떤 기호를 사용하여 표현할까?

- 우리 생활에서 쉽게 접할 수 있는 자판기 작동 알고리즘, 짝수/홀수 판별 알고리즘의 예를 순서도로 표현하여 설명한다.
- 순서도로 표현하는 방법이 자연어, 의사 코드로 표현하는 방법과 비교할 때 장단점이 무엇인지 설명한다.

## 참고자료

### 추가 선택 활동

• 교과서 103p  
지도서 155p

#### 거스름돈의 동전 수 최소화하기

물건을 사고 거스름돈을 받을 때 동전의 수를 최소화하기 위해서는 어떤 방법이 있을까?

**해결 아이디어:** 만약 거스름돈이 640원이라면, 500원짜리 동전 1개, 100원짜리 동전 1개, 10원짜리 동전 4개를 거슬러 받는 것이 거스름돈 640원에 대한 최소 동전의 수이다.

**문제 해결 방법 및 절차:** 거스름돈의 액수와 상관없이 최소의 거스름돈을 받기 위한 문제 해결 방법과 절차를 자연어, 의사 코드, 순서도 중 한 가지 방법을 선택하여 표현해 보자.



- 최소 동전 수의 문제 상황을 해결하기 위해서는 남은 거스름돈 액수를 넘지 않는 가장 큰 액면의 동전을 계속하여 선택하는 알고리즘을 다음과 같이 표현해 볼 수 있다.

#### 의사 코드(pseudo code)

- 의사(pseudo)는 '모조 의', '흉내 낸'이라는 의미이고, 코드(code)는 프로그래밍 코드를 말한다.
- 특정 프로그래밍 언어에 대한 지식이 없어도 알기 쉽게 표현이 가능하고, 다른 프로그래밍 언어로 전환하기 쉽다.

• 의사 코드를 사용할 경우 들여쓰기를 하면 이해하기 쉬워진다.

#### 순서도

1965년 10월 SC7 총회에서 표준안으로 결의된 기호를 사용한다. 순서도의 기호는 총 30개이며 기본 기호, 프로그래밍 관련 기호, 시스템 관련 기호 등 3가지 종류로 나뉜다. 기호와 그림을 사용하므로 알고리즘의 구조나 특성을 한 눈에 파악하기 쉽고 오류를 쉽게 찾을 수 있다.

#### 2 의사 코드

자연어의 표현 방법을 간략하게 구조화하여 표현한 것으로 프로그래밍 언어보다 이해하기 쉽다. 음료 자판기의 작동 알고리즘과 짝수·홀수를 판단하는 알고리즘을 의사 코드로 살펴보자.

##### 자판기의 작동 원리

- ① A ← 받은 동전
- ② A > 700  
예) B ← A - 700  
'B', '음료수' 출력  
아니오) A = 700?  
예) '음료수' 출력  
아니오) 'A' 출력



##### 짝수·홀수의 판별

- ① A ← 입력 값
- ② A ÷ 2의 나머지가 0?  
예) '짝수' 출력  
아니오) '홀수' 출력

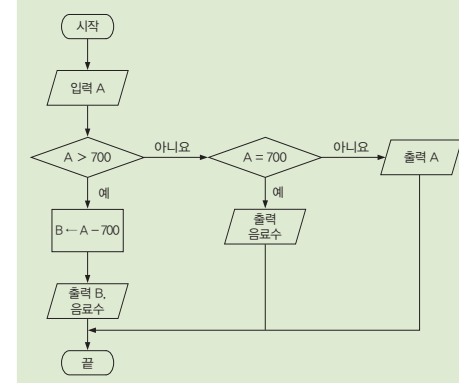
#### 3 순서도

순서도는 미리 약속된 기호나 그림 등을 이용해 논리적 절차, 흐름, 처리 방법 등을 표현하는 방법이다. 순서도는 다음과 같은 기호를 사용한다.

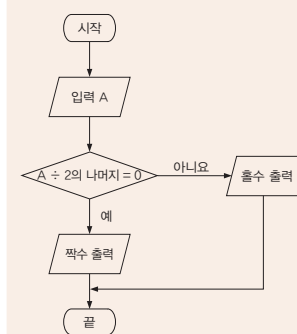
기호	명칭	의미
	단말 기호	알고리즘의 시작과 끝
	처리 기호	각종 연산이나 자료의 이동, 편집 등의 처리
	판단 기호	조건의 참, 거짓 등을 판단
	자료 입출력	자료의 입력 및 출력
	준비 기호	변수의 선언 및 초기값을 설정
	흐름선	실행 순서

음료 자판기의 작동 알고리즘과 짝수·홀수를 판단하는 알고리즘을 순서도로 살펴보자.

##### 자판기의 작동 원리



##### 짝수·홀수의 판별



알고리즘을 표현하는 설계는 그림을 그리기 전에 밑그림을 그리는 과정과 비슷하다. 알고리즘을 표현할 때는 전체의 흐름을 쉽게 이해할 수 있도록 흐름을 규칙적이고 단순하게 작성하는 것이 좋다.

알고리즘을 나타낼 때 실제로 해 보면 어느 하나의 표현 방법만 사용하는 것보다 세 가지를 함께 사용하여 알고리즘을 작성하는 것이 더 좋은 경우가 많다.

#### 2 알고리즘의 설계

문제를 해결하기 위한 방법과 절차를 구체적으로 표현하는 것을 알고리즘 설계라고 한다. 알고리즘을 표현하는 데 사용되는 설계 구조에는 순차 구조, 선택 구조, 반복 구조가 있다.

##### 1 순차 구조

순차 구조는 가장 간단하게 표현되는 구조로, 특별한 분기 또는 반복 없이 처음부터 끝까지 순서대로 처리하는 구조이다. 다음은 입력받은 두 수의 합을 순차 구조로 표현한 예이다.



III-2장 알고리즘 97

- 순서도를 실행한 결과를 같이 풀어 본 후 풀이 과정을 발표하게 하여 의견을 공유할 수 있도록 한다.
- 문제 해결 방법과 절차를 자연어, 순서도, 의사 코드로 다양하게 표현해 봄으로써 표현 방법을 비교해 본다.

- ① 거스름돈의 액수 change(= 640)를 입력받는다.
- ② 각 동전의 카운트를 n500 = n100 = n50 = n10 = n1 = 0으로 초기화한다(n500은 500원 동전의 카운트, n100은 100원 동전 카운트를 말한다).
- ③ change가 500보다 크거나 같으면 change ← change - 500(640 - 500 = 140)을 하고, 500원 동전의 카운트를 1 증가시킨다(n500 = 1). change가 500보다 작으면 반복 루프는 더 이상 수행하지 않는다.
- ④ 다음은 change가 100보다 크거나 같으면 change ← change - 100(140 - 100 = 40)을 하고, 100원 동전의 카운트를 1 증가시킨다(n100 = 1). change가 100보다 작으면 반복 루프는 더 이상 수행하지 않는다.

- ⑤ 다음은 change가 50보다 크거나 같으면 change ← change - 50을 하고, 50원 동전의 카운트를 1 증가시킨다. change가 50보다 작으면 반복 루프는 더 이상 수행하지 않는다.
- ⑥ 다음은 change가 10보다 크거나 같으면 change ← change - 10(40 - 10 = 30)을 하고, 10원 동전의 카운트를 1 증가시킨다(n10 = 1). change가 10보다 크거나 같은 동작이 3번 더 반복되므로 동전의 카운트(n10 = 4)가 된다. change가 10보다 작으면 반복 루프는 더 이상 수행하지 않는다.
- ⑦ 다음은 change가 1보다 크거나 같으면 change ← change - 1을 하고, 1원 동전의 카운트를 1 증가시킨다. change가 1보다 작으면 반복 루프는 더 이상 수행하지 않는다.
- ⑧ 동전 카운트들의 합을 리턴한다. n500 + n100 + n50 + n10 + n1 = 1 + 1 + 4 + 0 + 0 = 6이므로 6을 리턴한다.

### 추가 선택 활동

- 두 수의 차를 출력하는 알고리즘을 순서도로 표현해 본다.  
→ 두 수 A, B를 입력받아  
변수 A가 변수 B보다 크면 변수 A에서 변수 B를 빼고 그렇지 않으면 변수 B에서 변수 A를 뺀다.

순서도 표현	답
	㉠ : A > B ㉡ : R ← B - A

1 자연어로 표현하기

컨베이어 벨트로 이동하는 사과를 무게에 따라 3개의 상자로 나누는 장치의 작동 원리(자연어 표현)

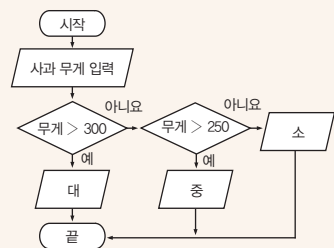
- 컨베이어 벨트로 사과가 이동하여 300g 초과는 '대' 등급 상자로 이동, 300~250g은 '중' 등급 상자로 이동, 나머지는 '소' 등급 상자로 이동시킨다.

2 의사 코드로 표현하기

- 1) 컨베이어 벨트로 사과 이동
- 2) 사과 무게 > 300g?

예) '대' 등급 상자로 이동  
아니요) 사과 무게 > 250g?  
예) '중' 등급 상자로 이동  
아니요) '소' 등급 상자로 이동

3 순서도로 표현하기



- 우리 생활에서 쉽게 접할 수 있는 자판기 작동 알고리즘, 짝수/홀수 판별 알고리즘의 예를 의사 코드로 표현하여 설명한다.
- 의사 코드로 표현하는 방법이 자연어, 순서도로 표현하는 방법과 비교할 때 장단점이 무엇인지 설명한다.

• 추가 선택 활동 •

- 선별기 활용 사례: 은행의 동전 분류기, 농산물 또는 과일 분류기, 폐기물 풍력 선별기 등이 있다.
- 국어, 수학, 정보 점수를 입력받아 총점과 평균을 계산하여 출력하는 과정을 의사 코드로 표현해 본다.



문제 해결 과정을 자연어·의사 코드·순서도로 표현하기

**문제 상황:** 컨베이어 벨트로 이동하는 사과를 무게에 따라 구분하여 3개의 상자로 나누는 장치의 작동 알고리즘을 만들어야 한다.  
컨베이어 벨트로 사과가 이동하여 300g 초과는 '대' 등급 상자로 이동, 300~250g은 '중' 등급 상자로 이동, 나머지는 '소' 등급 상자로 이동시킨다.



1 자연어로 표현하기

컨베이어 벨트로 사과 A가 이동한다.

2 의사 코드로 표현하기

의사 코드 표현

사과 A 입력

3 순서도로 표현하기

순서도 표현

시작

의사 코드 표현

```
INPUT 국어
INPUT 수학
INPUT 정보
SUM = 국어 + 수학 + 정보
AVG = SUM / 3
PRINT SUM
PRINT AVG
```

이와 같이 다른 프로그래밍 언어로 전환하기 쉽게 의사 코드를 표현하는 방법도 있다.

입력받은 두 수의 합을 출력하는 알고리즘

자연어	의사 코드	순서도
첫 번째 수를 입력받아서 A에 저장한다. 두 번째 수를 입력받아서 B에 저장한다. A의 값과 B의 값을 더하여 C에 저장한다. C의 값을 출력하고 끝낸다.	① A 입력 ② B 입력 ③ C ← A + B ④ C 출력	시작 ↓ INPUT A INPUT B ↓ C ← A + B ↓ PRINT C ↓ 끝

2 선택 구조

주어진 조건에 따라 처리하는 내용이나 순서를 다르게 진행하는 구조이다. 주어진 조건의 만족 여부에 따라 결과 값이 달라지며, 조건을 만족하면 “예(Yes)”의 흐름선을 따라 수행하고, 조건을 만족하지 않으면 “아니요(No)”의 흐름선을 따라 수행한다.

다음은 주민 등록 번호에서 성별을 판별하는 방법을 선택 구조로 표현한 예이다.

주민 등록 번호에서 성별 판별 알고리즘

의사 코드	순서도
① 주민 등록 번호 입력 ② A ← 주민 등록 번호 7번째 자리 숫자 ③ A = 1 or 3? 예) '남성' 출력 아니요) A = 2 or 4? 예) '여성' 출력 아니요) ①번으로 다시 입력 받기	시작 ↓ INPUT 주민 등록 번호 ↓ A ← 주민 등록 번호 7번째 자리 숫자 ↓ A = 1 or 3? 예 ↓ 남성 ↓ 끝 아니요 ↓ A = 2 or 4? 예 ↓ 여성 ↓ 끝

3 반복 구조

주어진 조건을 만족할 때까지 특정 작업을 반복하여 수행하는 구조를 말한다. 다음 예제와 같이 동일한 명령을 반복할 때에는 반복 구조를 사용하여 표현하면 간편하고 효율적이다.

• 순서도 속 연산자의 의미  
←: 오른쪽의 값을 왼쪽의 값에 넣는다.  
=: 오른쪽의 값과 왼쪽의 값이 같다.

핵·심·질·문

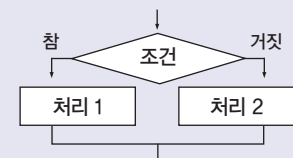
알고리즘을 표현하는 설계 구조에는 어떤 것이 있을까?

- 알고리즘을 설계하는 구조에는 순차 구조, 선택 구조, 반복 구조가 있음을 알게 하고, 일상생활에서 활용되고 있는 사례 중심으로 설명하여 이해력을 높인다.
- 순차 구조, 선택 구조를 설명할 때 자연어, 의사 코드, 순서도로 표현해 보도록 해 다양한 알고리즘 표현 방법을 접해 볼 수 있도록 한다.

- 순차 구조는 프로그램의 흐름이 순차적으로 위에서 아래로 또는 왼쪽에서 오른쪽으로 이루어지는 구조이며, 순차 구조의 처리 순서를 표현하면 다음과 같다.



- 선택 구조는 주어진 조건의 판단에 따라서 프로그램 흐름의 방향을 참과 거짓으로 분기하여 나가는 구조이며, 처리 순서를 표현하면 다음과 같다.



- 반복적인 순차 구조를 사용하면서 반복 구조의 필요성과 편리성을 알게 한다.

### 함께 해 보기 예시 답안

- A: 지원자가 맞는가?  
B: 받은 사람에게로 전할 내용을 말한다.  
C: 통화 내용을 말한다.

### 참/고/자/료

#### 추가 선택 활동

선택 구조를 실행하여 x, y의 출력 결과를 표현하면 다음과 같다.

선택 구조	풀이와 답
<pre> 시작 ↓ x ← 10 y ← 7 ↓ x &gt; y? ├─ 아니요 ─┤ y └─ 예 ─┘ x ↓ 끝 </pre>	<p><b>풀이</b> x ← 10, y ← 7일 때 x &gt; y 이므로 x값인 10을 출력한다.</p> <p><b>출력 결과</b> 10</p>

아래와 같은 프로그램을 작성하여 실행시키면 오른쪽과 같은 글자가 출력된다.

프로그램	화면 출력
화면 표시 "반갑다"	반갑다
화면 표시 "친구야"	친구야
화면 표시 "반갑다"	반갑다
화면 표시 "친구야"	친구야

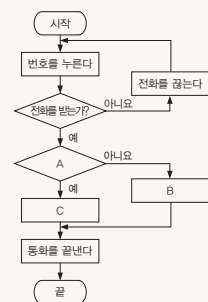
그러나 이렇게 두 가지 이상의 명령이 반복될 경우 다음과 같이 표현하면 좀 더 효율적으로 표현할 수 있다.

프로그램	화면 출력
반복하기 2번 { 화면 표시 "반갑다" 화면 표시 "친구야" }	반갑다 친구야 반갑다 친구야

예제 프로그램에서 '반복하기 2번 { }'은 { }안의 내용을 2번 반복해서 화면에 표시한다는 의미이다.

#### 문제 해결 과정 순서도 표현하기

승훈이 지원에게 집으로 전화를 걸어 용건을 말하는 순서는 아래와 같다. 문제 해결 과정을 순서도로 표현할 때 빈칸 A, B, C에 들어갈 적절한 말을 적으시오.



#### 순서

- 가. 전화번호를 누른다.  
나. 상대방이 전화를 받는지 확인한다.  
다. 통화가 되면 받은 사람이 지원이 맞는지를 확인한다.  
라. 지원이 맞으면 통화 내용을 말한다.  
마. 지원이 집에 없으면 받은 사람에게 전할 내용을 말한다.  
바. 상대방이 전화를 받지 않으면 전화를 끊는다.  
사. 통화를 끝낸다.

- A \_\_\_\_\_  
B \_\_\_\_\_  
C \_\_\_\_\_

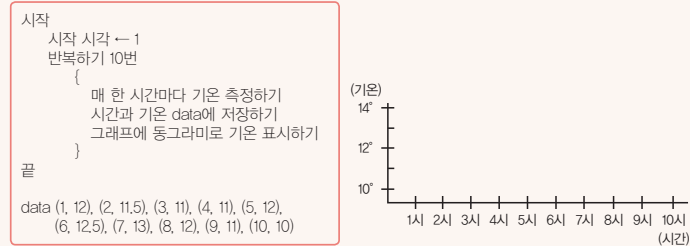


#### 반복 알고리즘 작성하기

1 화면에 아래와 같은 글자가 출력되도록 반복 구조로 표현하기 위해 \_\_\_\_\_을 채워 보자.

화면출력	프로그램
교통봉사 양로원봉사 교실청소 교통봉사 양로원봉사 교실청소 교통봉사 양로원봉사 교실청소	반복하기 _____번 { 화면 표시 "_____" 화면 표시 "_____" 화면 표시 "_____" }

2 가상 관측소에서 1시간마다 기온을 측정하는 알고리즘이다. 알고리즘을 보고 실행한 결과를 그래프로 나타내 보자.



#### 3 알고리즘의 활용

컴퓨터에서 자료를 저장하고 처리하기 위해서는 정렬의 과정이 필요하다. 정렬과 탐색은 자료를 구조화하고 결과를 처리하기 위해 매우 중요한 알고리즘이다.

정렬은 여러 개의 자료를 일정한 기준에 따라 순서대로 나열하는 것이다. 컴퓨터로 숫자를 정렬하기 위해 다양한 정렬 알고리즘을 사용한다. 그 가운데 버블 정렬(bubble sort)은 인접한 두 자료를 비교하여 가장 큰 데이터를 가장 뒤로 보내며 자리를 교환하는 방식이다.

탐색은 원하는 자료를 찾는 방법으로 순차 탐색과 이진 탐색이 있다.

정렬 알고리즘의 중요성  
탐색한 자료가 정렬되어 있지 않다면 순서대로 탐색하는 방법밖에 없다. 하지만 정렬되면 보다 다양한 탐색 방법이 사용될 수 있다.

### 핵심질문

알고리즘 설계 구조 중 반복 구조는 어떻게 표현할까?

- 반복 구조를 설명할 때 자연어, 의사 코드, 순서도로 표현해 보도록 해 다양한 알고리즘 표현 방법을 접해 볼 수 있도록 한다.
- 화면 출력과 같이 글자가 출력되기 위해서는 다음과 같이 프로그램을 작성해 볼 수 있다.

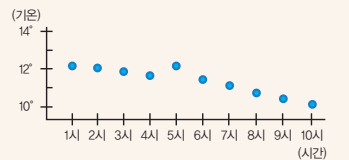
### 함께 해 보기 예시 답안

1

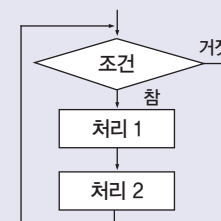
#### 프로그램

반복하기 3번  
{화면표시 "교통봉사"  
  화면표시 "양로원봉사"  
  화면표시 "교실청소" }

2



- 반복 구조는 특정 부분을 반복하여 처리하며 진행되는 구조이며, 일정 부분의 명령을 주어진 조건이 만족할 때까지 또는 주어진 조건이 만족하는 동안 반복 수행하는 형태이다. 반복 구조의 처리 순서를 표현하면 다음과 같다.





## 합계 해 보기 예시 답안

### 1 영어 단어 정렬(알파벳순)

정렬 전

1단계

2단계

3단계

4단계

5단계

정렬 완료

## 핵심질문

정렬과 탐색 알고리즘에는 어떤 것이 있을까?

정렬 알고리즘을 설명할 때 예제를 제시해 주거나 같이 실습을 해 봄으로써 이해도를 높인다.

다음은 버블 정렬을 이용하여 5개의 숫자를 오름차순으로 나타내는 방법이다.

### 5장의 숫자 카드를 버블 정렬 방법으로 오름차순으로 나타내기

단계	진행 과정	진행 내용
정렬되지 않은 상태	45 20 60 35 10	인접한 두 숫자를 비교하여 왼쪽의 숫자가 크면 자리 교환이 일어난다.
1단계	45 20 60 35 10	45, 20 중 45가 더 크므로 교환함.
	20 45 60 35 10	45, 60 중 60이 더 크므로 교환 안 함.
	20 45 60 35 10	60, 35 중 60이 더 크므로 교환함.
	20 45 35 60 10	60은 가장 큰 숫자로 고정됨.
2단계	20 45 35 10 60	20, 45 중 45가 더 크므로 교환 안 함.
	20 35 45 10 60	45, 35 중 45가 더 크므로 교환함.
	20 35 10 45 60	45는 두 번째 큰 숫자로 고정됨.
	20 35 10 45 60	20, 35 중 35가 더 크므로 교환 안 함.
3단계	20 10 35 45 60	35는 세 번째 큰 숫자로 고정됨.
	20 10 35 45 60	20, 10 중 20이 더 크므로 교환함.
4단계	10 20 35 45 60	오름차순 정렬이 완료됨.
	10 20 35 45 60	오름차순 정렬이 완료됨.

정렬 알고리즘은 데이터베이스, 인공지능 등의 다양한 분야에 적용되어 우리의 실생활 문제를 해결하며 세상을 변화시키는 데 중요한 역할을 하고 있다. 정렬 알고리즘 이외에 정보 통신 기술의 발달을 가져온 대표적인 알고리즘은 다음과 같다.



### 다양한 자료 정렬해 보기

다음과 같은 다양한 카드를 버블 정렬을 이용해서 정렬해 보자. 몇 번의 교환 과정이 필요한가?

#### 1 영어 단어 정렬(알파벳순)

create earth apple think computer binary

▲ 영어 카드

#### 2 여러 가지 음표 정렬(음표 길이순)

♪ ♪ ♫ ♬ ♧ ♨

▲ 음표 카드

지도서 148p

· 추가 선택 활동 · 참고

### 거스름돈의 동전 수 최소화하기

물건을 사고 거스름돈을 받을 때 동전의 수를 최소화하기 위해서는 어떤 방법이 있을까?



- 1 해결 아이디어: 만약 거스름돈이 640원이라면 500원짜리 동전 1개, 100원짜리 동전 1개, 10원짜리 동전 4개를 받으면 거스름돈 640원에 대한 최소 동전의 수이다.
- 2 문제 해결 방법 및 절차: 거스름돈의 액수와 상관없이 최소의 거스름돈을 받기 위한 문제 해결 방법과 절차를 자연어, 의사 코드, 순서도 중 한 가지 방법을 선택하여 표현해 보자.

5장의 숫자 카드, 음표 카드, 단어 카드 등을 버블 정렬의 방법을 이용하여 오름차순으로 나타내 보는 예제를 같이 풀어 본다.

## 합계 해 보기 예시 답안

### 2 여러 가지 음표 정렬(음표 길이순)

정렬 전

1단계

2단계

3단계

4단계

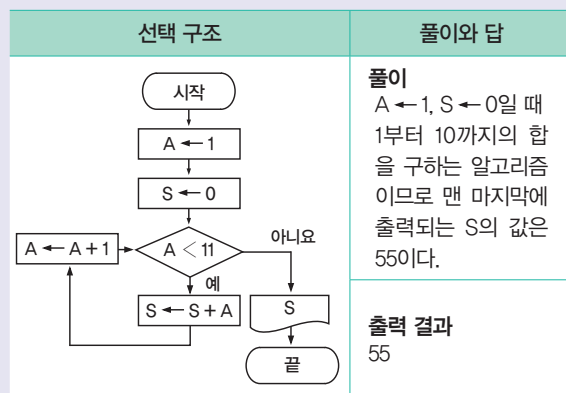
5단계

정렬 완료

## 참고자료

### · 추가 선택 활동 1 ·

1 다음 순서도를 실행한 결과 S의 값은 무엇일까?



### · 추가 선택 활동 2 ·

2 1번과 같은 결과가 어떻게 출력되는지 과정을 자연어로 표현해 보자.

#### 결과가 출력되는 과정

A ← 1, S ← 0에서부터 시작하여 1부터 10까지의 합을 차례대로 구한 다음, 맨 마지막 합을 출력한 후 알고리즘을 끝낸다.

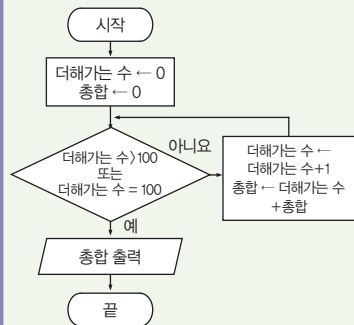
- **정렬:** 여러 개의 자료를 일정한 기준에 따라 순서대로 나열하는 것을 말하며, 정렬을 하면 자료를 보기 좋게 정리할 수 있고 원하는 자료를 쉽게 찾아낼 수 있다.
- **버블 정렬:** 인접한 두 자료를 비교하여 자리를 교환하는 방법이다.
- **삽입 정렬:** 정렬되지 않은 데이터를 이미 정렬된 자료의 알맞은 위치에 삽입하는 방법이다.
- **선택 정렬:** 자료의 첫 번째 위치를 기준 위치로 정하고, 나머지 자료를 기준 위치에 있는 자료와 비교하여 교환하거나 교환하지 않는 방법이다.

- **퀵 정렬:** 기준 키를 기준으로 작거나 같은 값을 지닌 데이터는 앞으로, 큰 값을 지닌 데이터는 뒤로 가도록 하여 작은 값을 갖는 데이터와 큰 값을 갖는 데이터로 분리해 가며 정렬하는 방법이다.
- **셸 정렬:** 주어진 입력 파일을 매개 변수의 값에 따라 여러 개의 부 파일로 나누고, 각 부 파일은 삽입 정렬법으로 정렬하는 과정을 되풀이하는 방법이다.

# 실습 활동

## 실습 활동 답안

### 1 순서도



### 2 의사 코드

- ① 국어, 수학, 영어 점수 입력하기
- ② 총점 ← 국어+수학+영어
- ③ 평균 ← 총점 / 3
- ④ 만약 평균 ≥ 60?
- ⑤ 예) "합격입니다." 출력
- ⑥ 아니요) "불합격입니다." 출력

3 (가) 대출 안됨 [반납일이 지난 책을 가지고 있으므로]

(나) 14일 대출 [제한이 표시되지 않은 책을 선생님이 대출하므로]

(다) 7일 대출 [제한이 표시되지 않은 책을 학생이 대출하므로]

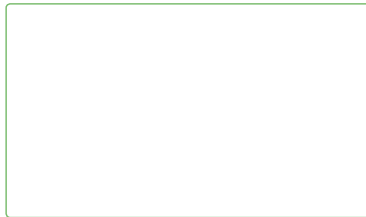
(라) 1일 대출 [제한이 표시된 책이므로]

## 실습 활동

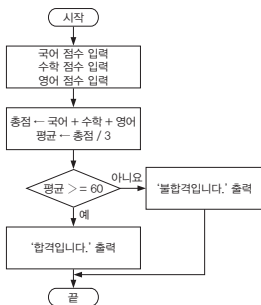
1 다음은 1부터 100까지 합을 구하는 의사코드이다. 이것을 순서도로 표현해 보자.

시작  
더해가는 수 ← 0  
총합 ← 0  
더해가는 수 > 100 까지 반복  
{  
더해가는 수 ← 더해가는 수 + 1  
총합 ← 더해가는 수 + 총합  
}  
총합 출력  
끝

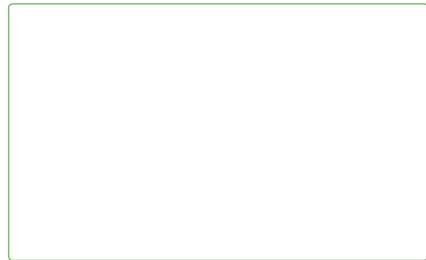
### 순서도



2 다음 그림은 세 과목의 점수를 입력받아 총점, 평균을 구한 뒤 합격 여부를 출력하는 순서도이다. 의사 코드로 표현해 보자.



### 의사 코드

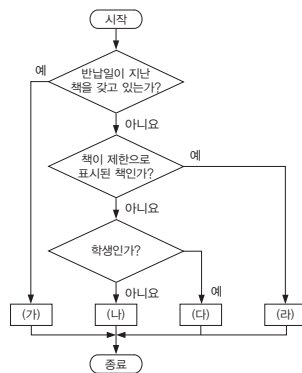


3 우리 학교 도서관은 아래와 같은 도서 대출 규칙이 있다. 도서 대출 규칙에 따라 순서도를 표현할 때 (가), (나), (다), (라)에 들어갈 내용은 무엇인가?

### 도서 대출 규칙

- ▶ '제한'으로 표시된 책은 누구나 대출 기간이 1일이다.
- ▶ '제한'이 표시되지 않은 책은 대출 기간이 선생님 14일, 학생 7일이다.
- ▶ 반납일이 지난 책을 가지고 있는 사람은 대출이 안 된다.

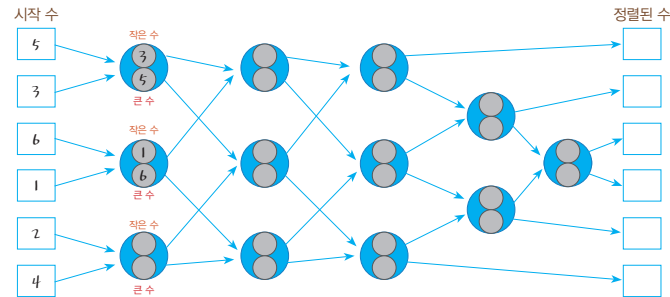
- (가) \_\_\_\_\_
- (나) \_\_\_\_\_
- (다) \_\_\_\_\_
- (라) \_\_\_\_\_



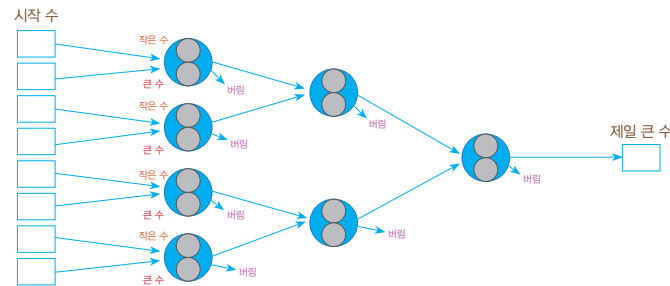
104

Ⅲ부 문제 해결과 프로그래밍

4 아래의 그림과 같이 숫자를 정렬하는 망(정렬망)을 이용하여 숫자를 정렬해 보자. 이웃한 두 수를 비교하여 더 큰 수는 위쪽 화살표를 따라가고 작은 수는 아래쪽 화살표를 따라가며, 원 안에는 작은 수를 위로, 큰 수를 아래 기재한다.



5아래 그림은 제일 큰 수를 찾기 위한 정렬망이다. 4번과 같은 규칙으로 큰 수가 위쪽 화살표를, 작은 수가 아래쪽 화살표를 따라가지만 원 안에는 큰 수를 위로 기재하여 여러분이 시작 숫자를 넣어서 제일 큰 수를 찾아보자.



## 자기 평가

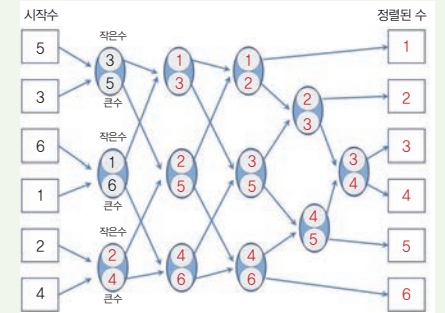
평가 내용	잘함	보통	노력 요함
1. 자연어, 순서도, 의사 코드를 활용하여 알고리즘을 표현할 수 있다.			
2. 순차, 선택, 반복 구조를 활용하여 알고리즘을 설계할 수 있다.			
3. 다양한 정렬 알고리즘과 다양한 분야에 활용되는 알고리즘을 이해할 수 있다.			

Ⅲ-2장 알고리즘

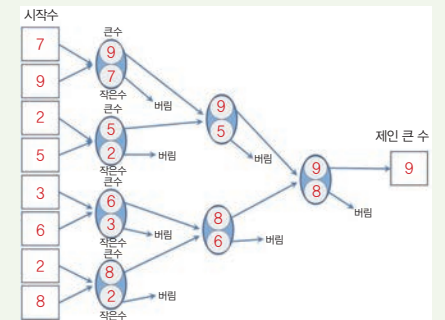
105

## 실습 활동 답안

### 4



### 5



## 추가 선택 활동

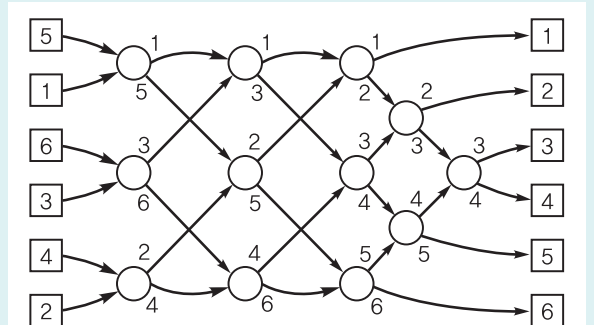
### 정렬망 활동

이 활동은 어떻게 컴퓨터가 정렬 네트워크를 사용하여 임의의 수들을 정렬하는지 보여준다.

- ① 여섯 명이 한 그룹을 만든다. 그리고 한 번에 한 그룹만이 이 네트워크를 사용한다.
- ② 각 팀의 구성원들은 숫자가 쓰인 카드를 가져간다.
- ③ 구성원들은 네트워크 왼쪽에 있는 네모 칸에 서 있다. 학생들이 가지고 있는 숫자들은 뒤죽박죽 순서가 될 것이다.
- ④ 표시된 선을 따라 이동한 후 원에 도착했을 때 학생들은 다른 곳에서 오는 두 군가를 반드시 기다려야 한다.

5 다른 멤버가 자신의 원에 도착했을 때 그 두 명의 카드에 적힌 숫자를 비교해 본다. 더 작은 수의 카드를 가진 사람이 왼쪽에 있는 선으로 이동한다. 만약 더 높은 수를 가지고 있다면 오른쪽 선으로 이동한다.

6 네트워크의 오른쪽에 모든 멤버가 도착했을 때, 학생들이 든 카드는 크기 순으로 정렬이 되었을 것이다. 만약 한 명에서 오류가 생겼다면 그 학생은 다시 시작해야 한다. 작은 값은 왼쪽으로 가고 큰 값은 오른쪽으로 가는 네트워크 마디(선)와 노드(원)의 작동을 이해했는지 확인해 본다.



## 읽기 자료 & 토론허기

- 수많은 자료 중에 찾고자 하는 단어를 쉽게 검색하기 위해 인덱스를 활용하는 예를 들어 설명하면서 이해도를 높인다.
- 토론허기에서는 문제를 해결하기 위한 최선의 알고리즘을 발견하였는데, 해답을 찾는 데 100년이 걸린다고 할 때 최고의 알고리즘이라고 말할 수 있는지를 생각해 보는 문제이니 모둠원과 충분한 토론허를 거치고 이야기 나눌 수 있는 시간을 부여 한다.

### 읽기 자료

#### 검색 엔진과 색인

검색 엔진은 수많은 자료 중에 찾고자 하는 단어를 쉽게 검색하기 위해 색인은 사용한다. 검색 엔진에서 검색을 빠르게 하는 아이디어는 색인이 페이지 번호뿐만 아니라 페이지 안의 위치도 저장해야 한다는 생각에서 출발한다. 아래 그림의 검색 단어 번호는 세 페이지짜리 자료 모음에 단어 위치를 추가적으로 작성한 예시이다. 여기서 위치는 페이지 내에 단어의 위치를 말한다. 예를 들면 2번째 단어의 위치는 2, 19 번째 단어의 위치는 19로 표시한다.

그림의 색인 표는 페이지 번호와 단어 위치를 모두 저장한 곳에서 나온 색인 번호들이다. 이러한 방법으로 검색하려는 단어를 입력하면 검색 엔진은 색인 표에 입력된 수치를 이용하여 정확하고 빠르게 검색 작업을 처리하게 된다.

검색 단어 번호					색인 표	
1 페이지	the 1	cat 2	sat 3	on 4	a	3-5
2 페이지	the 1	dog 2	stood 3		cat	1-2, 3-2
3 페이지	while 4	a 5	dog 6	sat 7	dog	2-2, 3-6
					mat	1-6, 2-6
					on	1-4, 2-4
					sat	1-3, 3-7
					stood	2-3, 3-3
					the	1-1, 1-5, 2-1, 2-5, 3-1
					while	3-4

색인의 첫 줄에 'a'란 단어가 데이터 모음에서 정확히 한 번 등장하고 3페이지의 다섯 번째 단어 뜻이다. 이러한 방법은 문장에 대해 질문한 것을 효율적으로 처리하기 위해 사용한다.

예를 들어 'cat sat'을 찾고자 할 때 먼저 개별 단어의 위치를 찾는 색인에서 추출한다. 그러면 'cat'에 대해서는 1-2와 3-2를, 'sat'에 대해서는 1-3과 3-7을 얻는다. 두 단어의 위치를 추출했지만 두 단어가 정확한 순서로 붙어 있지 않을 수 있다. 다행히도 위치 정보로부터 이를 확인할 수 있다. 1페이지를 살펴 보면 'cat'이 위치 2에 있고, 'sat'이 위치 3에 있으므로 'sat'이 'cat' 바로 다음에 있다는 사실을 알 수 있다. 따라서 우리가 찾고자 하는 'cat sat'은 1페이지의 위치 2에서 시작함에 틀림없으며, 이렇게 간단한 방법이 검색 엔진을 작동하는 열쇠가 되었다.

출처: '미래를 바꾼 9가지 알고리즘', 존 매코믹

### 토론허기

내비게이션과 같이 실생활에 활용된 다양한 알고리즘을 찾아보고 우리 생활에 미친 영향에 대해 논해 보자.

106

III 부 문제 해결과 프로그래밍

## 참고자료

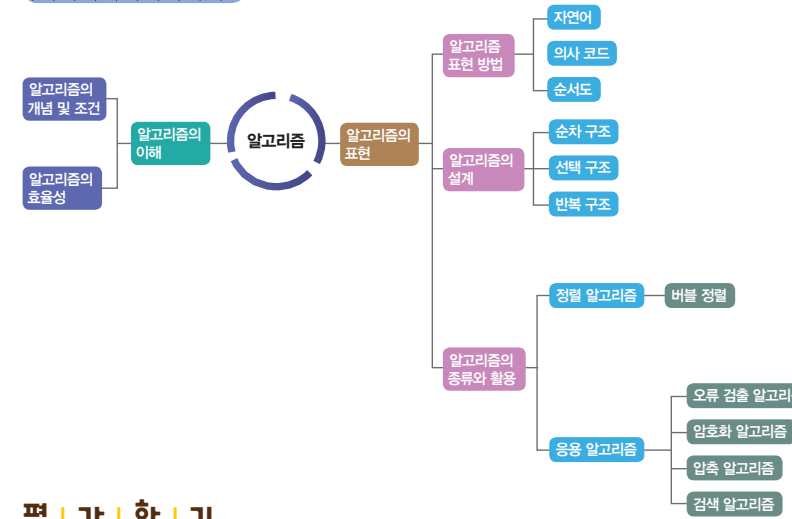
### 알고리즘의 복잡성

알고리즘은 보통 최선, 최악, 평균의 3가지 복잡성 중 하나에 속한다. 최선의 경우는 입력 값이 주어졌을 때 알고리즘이 처리하는 횟수를, 최대한 줄이는 것이다. 이러한 성능을 고려하는 것을 '시간 복잡성'이라고 한다. 또한 알고리즘이 수행될 때 얼마나 많은 공간이 필요한가를 고려하는 것을 '공간 복잡성'이라고 한다.

시간 복잡도와 공간 복잡도 사이에는 상반되는 관계가 있을 때가 많은데, 처리 속도가 빠른 알고리즘은 생각한 것보다 더 많은 저장 공간을 사용할 것이고, 적은 메모리 공간을 사용하는 알고리즘은 많은 공간을 사용하는 알고리즘보다 처리 속도가 느릴 수 있다.

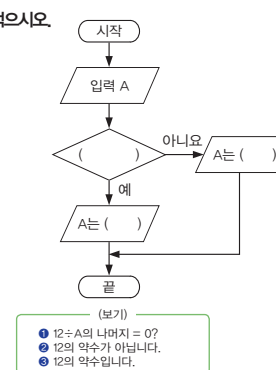
따라서 알고리즘을 작성할 때에는 시간 복잡성과 공간 복잡성 모두를 고려해야 한다.

### 배운 내용 정리하기

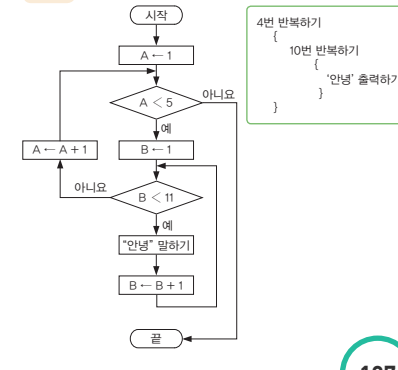


### 평가하기

- 문제를 해결하기 위한 방법을 순서대로 나열한 절차를 무엇이라고 할까?
- 다음은 입력 A가 12의 약수인지 판별하는 알고리즘을 구현한 순서도이다. 빈칸에 알맞은 내용을 보기에서 찾아 번호를 적으시오.
- 다음 알고리즘을 실행하면 '안녕'을 몇 번 말할까?



(보기)  
 ① 12 ÷ A의 나머지가 0?  
 ② 12의 약수가 아닙니다.  
 ③ 12의 약수입니다.



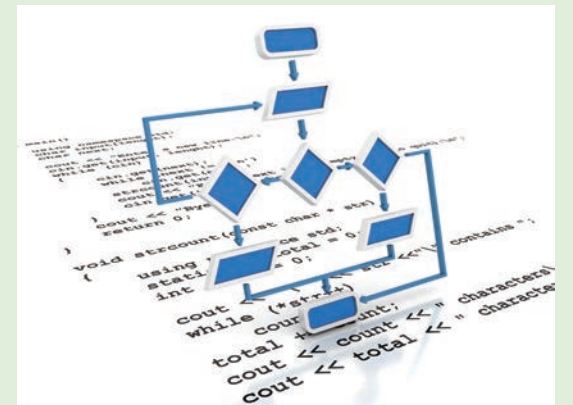
III-2장 알고리즘

107

## 배운 내용 정리 & 평가하기

### 평가하기 답안

- 알고리즘
- ① 12/A의 나머지 = 0?  
 A는 ( ③ ) (예) A는 12의 약수입니다.  
 A는 ( ② ) (아니요) A는 12의 약수가 아닙니다.
- ③ 40 [안녕]이라는 글자를 10번씩 4번 반복하므로 총 40번 말한다.]



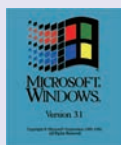


## 핵심질문

프로그래밍 언어란 무엇일까? 실생활 문제를 해결하기 위한 프로그램은 어떻게 만들 수 있을까?

- 동기 유발 만화를 살펴보고 프로그램 개발의 필요성에 대해 이야기해 본다.
- 진로 교육과 연계되는 방향으로 수업을 진행한다. 프로그래밍과 관련된 직업에는 무엇이 있는지 조사해 볼 수 있다.
- 소개된 프로그램 개발자들의 활동 내용을 보여 줄 수 있는 참고 자료를 제시한다(하단 자료 참고).

### 프로그램 개발자들의 활동 내용



빌 게이츠  
마이크로소프트 창업



스티브 잡스  
애플 창업



마크 저커버그  
페이스북 창업



드류 휴스턴  
드롭박스 창업

출처: 구글 이미지

▶ 프로그램 개발자 관련 영상: [https://youtu.be/SZK4ic4IF\\_I](https://youtu.be/SZK4ic4IF_I)

빌 게이츠, 마크 저커버그, 드류 휴스턴 “놀라운 힘”

이 나라 모든 사람이 컴퓨터 프로그래밍을 배워야 하는 이유는?

## 학습 목표

- 프로그래밍 언어를 이해하고 자료의 입력과 출력을 할 수 있다.
- 변수를 이해하고, 변수와 연산자를 사용하여 프로그램을 작성할 수 있다.
- 순차, 선택, 반복 구조의 명령을 사용하여 구조적인 프로그램을 작성할 수 있다.
- 실생활의 문제를 해결하기 위한 프로젝트를 설계하고 프로그램으로 구현할 수 있다.

### 프로그래머(programmer)는 어떤 직업?

선희는 스마트폰 앱을 만드는 개발자가 되고 싶고, 재영은 온라인 가상 게임을 만들거나 영화에 들어가는 세밀한 3D 영상 프로그래머가 되길 원한다. 그래서 직업 정보 사이트를 통해 프로그래머라는 직업에 대해 조사하였다.

“컴퓨터 프로그래머는 컴퓨터가 인식할 수 있는 언어를 사용해서 필요한 명령 체계나 시스템을 만드는 일을 하는 사람을 말한다.”

프로그래머는 다양한 언어로 다양한 환경에서 프로그래밍을 하는 사람으로 세분화되어 있다.

프로그래머들은 모바일 앱이나 게임 프로그래머, 운영 체제나 네트워크 프로그램을 다루는 시스템 엔지니어 외에도 빅 데이터 분석가, IT 컨설턴트 등으로 직업이 다양화되는 경우도 있으며, 프로그래머들의 정책을 결정하는 최고 기술 경영자(CTO: Chief Technical Officer)가 되어 기술적으로 중요한 회사의 방향을 결정하는 책임자가 될 수도 있다.



유명한 IT 기업 수장들과의 프로그래밍에 대한 인터뷰 내용을 살펴보자.



## 생각해보기

위의 만화를 보고 프로그래머가 되려면 어떤일부터 시작하면 좋을지 생각해 보자.

### IT 관련 직업의 종류

직업명	하는 일
프로그래머	프로그래밍 언어를 통해서 프로그램을 개발하는 직업
서버 엔지니어	운영 체제(윈도, 리눅스 등) 서버를 관리·유지·보수하는 직업
네트워크 엔지니어	네트워크에 대한 운영과 관리, 유지·보수 업무를 수행하는 직업
데이터베이스 엔지니어	데이터를 관리하며 저장, 유지·보수 업무를 수행하는 직업
정보 보안 전문가	외부 공격으로부터 해킹을 방어하며 취약점을 분석하여 보안하는 업무를 수행하는 직업

핵·심·질·문

사용할 프로그래밍 언어의 개발 환경 및 특성을 이해하고 있는가?

- 주변에서 사용하는 애플리케이션을 비롯한 각종 프로그램에는 무엇이 있는지 이야  
기해 보고, 그것을 만들기 위해 프로그래밍  
언어가 존재하는 것을 알 수 있게 한다.
- 프로그래밍과 관련된 여러 가지 용어를  
구분하여 이해할 수 있게 한다. 또한 다양  
한 고급 언어를 소개한다(하단 자료 참고).

참/고/자/료

· 다양한 고급 프로그래밍 언어

■ C언어

C는 1972년 당시 새로 개발된 유닉스 운영 체제에서 사용하기 위해 개발한 프로그래밍 언어이다. 유닉스 시스템의 바탕 프로그램은 모두 C로 쓰였고, 많은 운영 체제의 커널도 C로 만들어졌다. 오늘날 많이 쓰이는 C++은 C에서 객체 지향형 언어로 발전된 것이다.

■ 자바

자바는 썬마이크로시스템즈에서 개발한 객체 지향적 프로그래밍 언어이며, 처음에는 가전제품 내에 탑재해 동작하는 프로그램을 위해 개발했지만 현재 웹 애플리케이션 개발에 가장 많이 사용하는 언어 가운데 하나이고, 모바일 기기용 소프트웨어 개발에도 널리 사용하고 있다.

■ PHP

PHP(Hypertext Preprocessor, 하이퍼텍스트 프리프로세서)는 프로그래밍 언어의 일종이다. 원래는 동적 웹 페이지를 만들기 위해 설계되었으며 이를 구현하기 위해 PHP로 작성된 코드를 HTML 소스 문서 안에 넣으면 PHP 처리 기능이 있는 웹 서버에서 해당 코드를 인식하여 작성자가 원하는 웹 페이지를 생성한다. 근래에는 PHP 코드와 HTML을 별도 파일로 분리하여 작성하는 경우가 일반적이다.

■ 파이선

파이선(Python)은 프로그래머인 귀도 반 로섬이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체 지향적, 동적 타이핑(dynamically typed) 대화형 언어이다. 파이선은 비영리의 파이선 소프트웨어



3-1 프로그래밍

사람들은 대화를 하거나 글을 쓸 때 각 나라의 언어를 사용한다. 내가 컴퓨터와 대화하거나 소셜 네트워크 서비스(SNS) 애플리케이션, 문서 작성 프로그램 등을 만들려면 프로그래밍 언어를 사용한다. 프로그래밍 언어를 살펴보면서 각 언어의 특징을 알아보자.

1 프로그래밍의 이해

우리가 일상생활에서 사용하는 인터넷 웹 브라우저, 스프레드시트, 스마트 앱을 프로그램이라 한다. 프로그램(program)이란 프로그래밍 언어를 통해 컴퓨터가 사용자의 명령을 알아들을 수 있도록 작업을 지시하고, 동작시키는 명령어의 모임이다. 프로그래밍 언어(programming language)는 컴퓨터와 소통할 수 있도록 문법을 갖춘 기호 체계이다. 프로그래밍 언어를 이용하면 문제의 흐름이나 동작 방법을 지시하고 전달할 수 있다.

이 프로그램을 작성하는 일을 프로그래밍(programming)이라고 하고, 이 프로그래밍을 통해 소프트웨어를 만드는 사람을 프로그래머(programmer)라고 한다.

2 프로그래밍 언어의 유형

1 프로그래밍 언어

프로그래밍 언어는 문자와 숫자, 기호로 이루어져 있다. 이때 프로그램을 표현하는 언어에 대해서 인간이 이해하기 쉽고 인간의 사고에 적합한 개념과 구조를 갖느냐에 따라 저급 언어와 고급 언어로 구분한다.

저급 언어	고급 언어
사람보다는 기계가 이해하기 쉽게 작성된 컴퓨터 중심의 프로그래밍 언어이다. 프로그램의 처리 속도는 빠르지만 사용자가 프로그램을 작성하는 것은 어렵다.	사용자가 이해하기 쉽도록 사람들이 사용하는 자연어와 비슷한 구조를 가지고 있다. 프로그램 작성은 쉽지만 처리 속도가 저급 언어보다 느리다.
기계어와 어셈블리 언어	C언어, 자바(Java), PHP 등

- 기계어  
컴퓨터가 직접 이해할 수 있는 0과 1로 구성된 컴퓨터 프로그래밍 언어이다.
- 어셈블리 언어  
기계어와 일대일로 대응시켜 기호화한 언어로, 기계어보다 쉽게 프로그래밍할 수 있다.

어 재단이 관리하는 개방형, 공동체 기반 개발 모델이다. C언어로 구현된 C파이션 구현이 사실상의 표준이다.

■ 자바스크립트

자바스크립트(JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용하며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 사이드 네트워크 프로그래밍에도 사용되고 있다.

■ 베이식

베이식(BASIC: Beginner's All-purpose Symbolic Instruction Code)은 프로그래밍 언어의 하나이다. 절차형 언어로, 교육용으로 개발되어 언어의 문법이 쉬우며, 최초 제품 이래로 인터프리터 방식이 많았으나 최근에는 컴파일러 방식도 많다. 현재는 다양한 종류의 베이식이 존재하며 문법도 서로 많이 차이가 난다. 서로 다른 종류 사이의 소스 코드는 호환되지 않는다.

출처: 위키피디아

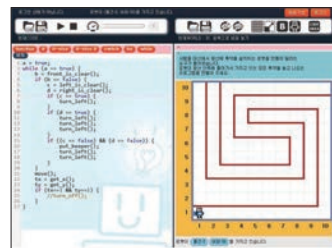
언어 번역 프로그램은 어셈블리 언어나 고급 언어와 같이 사람 중심의 언어로 작성된 프로그램을 컴퓨터가 직접 이해할 수 있는 기계어로 번역해 주는 프로그램이다. 어셈블러, 인터프리터, 컴파일러 등이 있다.



2 교육용 프로그래밍 언어

교육용 프로그래밍 언어(EPL: Educational Programming Language)는 학생들이 쉽게 프로그래밍을 배울 수 있도록 손쉬운 인터페이스와 사용법을 갖추고 있다.

교육용 프로그래밍 언어에는 문자로 명령어를 직접 입력하는 로고, 러플, 플레이봇, 파이선과 같은 텍스트 기반의 프로그래밍 언어와 명령 블록을 이용하는 스크래치, 엔트리, 앨리스, 스넵 등과 같은 블록 기반 프로그래밍 언어로 구분된다. 교육용 프로그래밍 언어는 실제적인 문제를 직접 구현하여 해결하려는 것이 아니라 컴퓨팅 사고력, 문제 해결력을 신장시키기 위한 것이다.



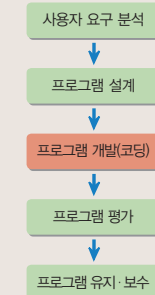
▲ 텍스트 기반의 교육용 프로그래밍 언어(플레이봇)



▲ 블록 기반의 교육용 프로그래밍 언어(스크래치)

- 어셈블러(Assembler)  
어셈블리 언어로 작성된 프로그램을 실행 가능한 기계어 형태로 바꾸어 주는 프로그램이다.
- 인터프리터(interpreter), 컴파일러(compiler)  
고급 언어로 작성된 프로그램을 즉시 실행 가능한 기계어로 바꿔 주는 프로그램이다.

● 프로그램 개발 과정



핵·심·질·문

교육용 프로그래밍 언어란 무엇일까?

- 일반 프로그래밍 언어와 교육용 프로그래밍 언어의 인터페이스를 비교할 수 있도록 자료를 제시한다.
- 교육용 프로그래밍 언어에는 어떠한 것들이 있는지 소개한다(하단 자료 참고).
- 실제로 블록 기반의 교육용 프로그래밍 언어를 체험해 볼 수 있는 시간을 가진다.

밍 언어이다. 로고는 리스프 언어의 개작이자 방언으로, 일부 사람들은 괄호 없는 리스프라 부르기도 한다. 오늘날 로고는 리스트, 파일, I/O, 재귀적 용법을 다룰 수 있는 기능들이 있다.

■ 엔트리

엔트리는 3D 스토밍으로 프로그래머가 아닌 평범한 사람도 이해하고 사용할 수 있게 만든 프로그래밍 언어이다.

■ 스쿼크 이토이

어린이의 컴퓨터 환경 및 교육에 사용되는 것을 상정한 객체 지향 프로토타입 기반 프로그래밍 언어이다. 스쿼크(Squeak)는 애플이 1996년에 개발한 애플컴퓨터의 스몰토크 80(Smalltalk-80)에서 파생된 객체 지향 클래스 페이스, 리플렉티브의 Smalltalk 구현이다. Etoys는 스쿼크 같은 다른 스쿼크(Squeak)형 교육용 프로그래밍 환경에 영향을 주었다.

■ 플레이봇

코딩 교육 도구로 널리 활용되고 있는 스크래치, 엔트리 등의 블록형 도구들이 중학생, 고등학생을 대상으로 하는 문제 해결 교육에는 부족해 한계를 느낀 한 고등학교 교사가 직접 개발한 교육도구이다. '러플(Rur-ple)'이란 코딩 교육 도구를 참고해 만들어졌다. 러플은 파이션 학습을 위한 텍스트형 코딩 교육 도구다.

출처: ZDNet Korea

■ 코두

코두(Kodu)는 2009년 발표된 마이크로소프트 엑스박스(XBOX)용 게임 개발 도구이다. 프로그래밍과 논리적 사고를 만 8세 정도의 아이들에게 가르칠 수 있을 정도로 혁신적인 비주얼 인터페이스이다.

출처: 코두(Kodu Game Lab): 비주얼 프로그래밍 <http://erke2000.blog.me/220091413668>

출처: 위키피디아



1 (순서대로) (코딩), 원시 프로그램, (컴파일), 목적 프로그램, (링크)

2 지도서 162~163쪽 '다양한 고급 프로그래밍 언어', '교육용 프로그래밍 언어' 참고 예 자바

- 객체 지향적 프로그래밍 언어
- 처음에는 가전제품 내에 탑재해 동작하는 프로그램을 위해 개발했지만 현재는 웹 애플리케이션 개발에 가장 많이 사용하고 있다.
- 모바일 기기용 소프트웨어 개발에도 널리 사용하고 있다.

### 핵·심·질·문

프로그래밍은 어떠한 과정으로 이루어질까? 프로그래밍 언어에는 무엇이 있을까?

- 프로그래밍의 패러다임 변화를 소개할 수 있다(하단 자료 참고).

### 참/고/자/료

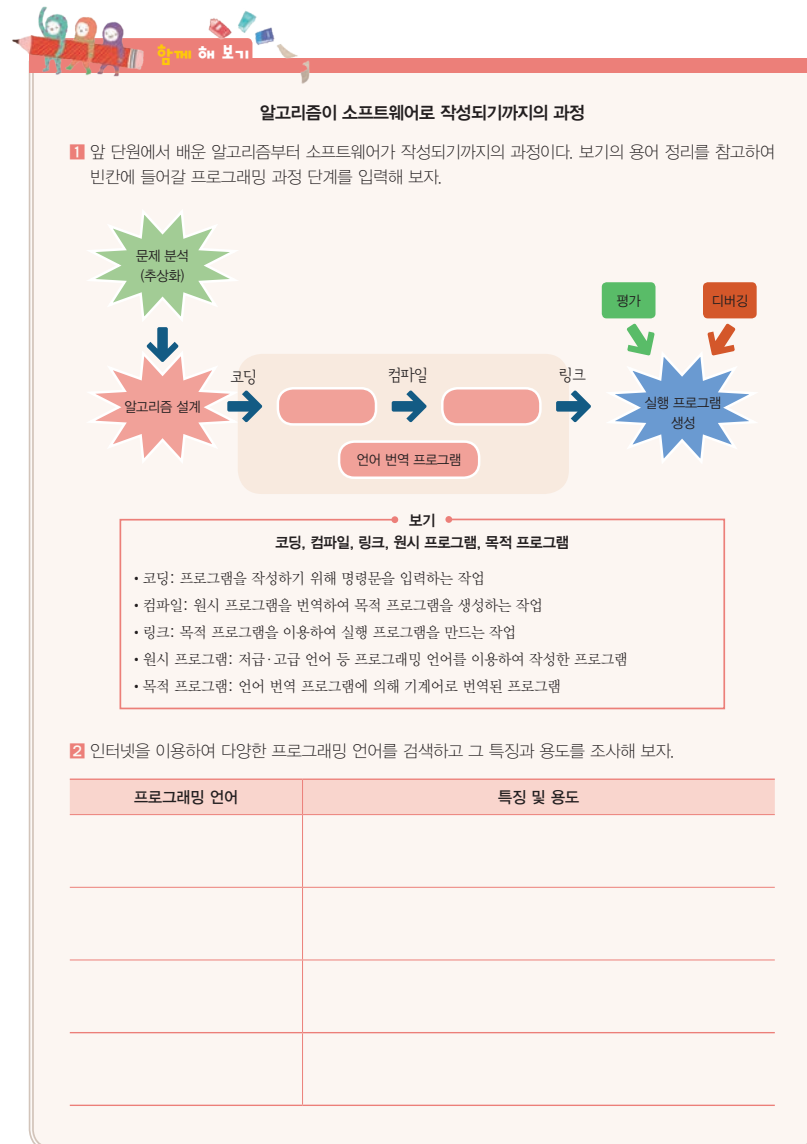
- 프로그래밍의 패러다임 변화

#### 1 절차형

명령형 프로그래밍과 동의어로 쓰이기도 하지만, 프로시저 호출의 개념을 바탕으로 하고 있는 프로그래밍 패러다임을 의미하기도 한다. 복잡도가 지나치지 않고 유지·보수가 쉬워 단순한 순차적 프로그래밍이나 비구조적 프로그래밍보다 여러 상황에서 장점이 많다.

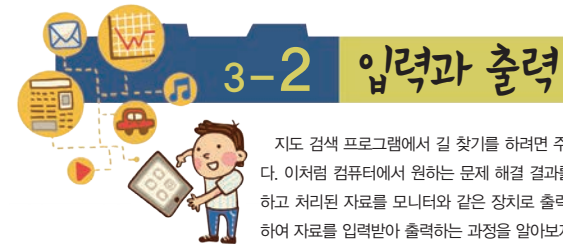
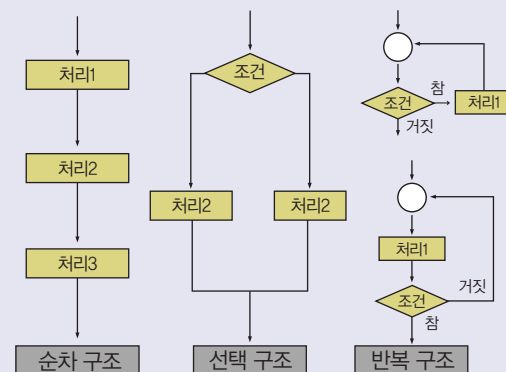
```

10 Input A
15 Input B
20 B = A + 10
30 IF B > 12 GOTO 60
40 C = B / 3
50 IF C < 24 GOTO 10
60 Write C
70 IF Write Failed GOTO 15
80 Input D
    
```



#### 2 구조형

절차적 프로그래밍의 하위 개념으로 볼 수 있다. GOTO문을 없애거나 GOTO문에 대한 의존성을 줄여 주는 것으로 가장 유명하다.



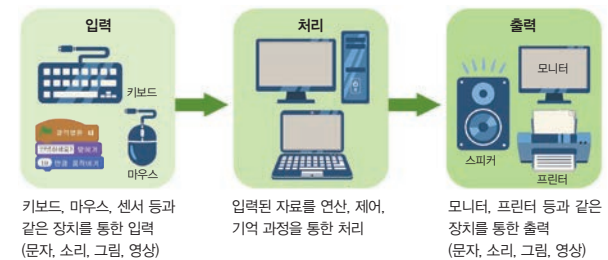
#### 1 프로그래밍 언어의 입력과 출력

자동차 판매기나 교통 카드의 동작과 같이 일상생활 속에서도 다양한 입력과 출력이 존재한다. 컴퓨터에서도 문제 해결을 위해서는 처리할 자료를 입력받아 처리 과정을 거친 후 해결된 실행 결과를 모니터나 파일 등으로 출력하게 된다.

입력과 출력 자료 형태는 숫자, 문자, 문자열뿐만 아니라 소리, 그림, 동영상 등 다양한 형태로 존재한다. 그러므로 프로그램의 목적에 따라 적합한 자료를 적절히 이용할 수 있도록 해야 한다.

모니터 화면에 나타난 고양이와 '안녕하세요?' 하는 인사를 출력하게 하려면 어떤 자료를 입력하여야 할까?

교육용 프로그래밍 언어를 활용하여 스프라이트를 동작시키기 위해서는 다양한 명령 블록을 입력하고 실행하여 처리 결과를 모니터와 같은 장치로 출력해야 한다.



### 3-2

## 입력과 출력

#### 핵·심·질·문

다양한 형태의 자료를 입력받아 처리하고 출력하기 위한 프로그램을 작성할 수 있는가?

- 실제로 지도 검색 프로그램에서 길 찾기를 해 보면서 입력과 출력에 대해 알아본다.
- 사람의 정보 처리 과정과 컴퓨터의 입출력을 비교하여 설명한다(하단 자료 참고).

- 사람의 정보 처리 과정과 컴퓨터의 입출력 비교

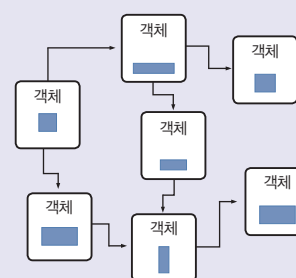


입력	문제를 읽어 들인다.	사람: 눈 컴퓨터: 입력 장치(자판)
처리	문제를 기억하고, 이를 계산하여 값을 다시 기억시킨다. 이러한 입력, 기억, 연산, 제어, 출력 등 각 기능이 순서대로 이루어지도록 한다.	사람: 두뇌 컴퓨터: 주기억 장치, 중앙 처리 장치(연산·제어 장치)
출력	처리 결과를 표현한다.	사람: 입(말), 손(글씨) 컴퓨터: 출력 장치(모니터, 프린터)

#### 3 객체 지향형

객체 지향 프로그래밍은 컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 '객체'들의 모임으로 파악하고자 하는 것이다. 각각의 객체는 메시지를 주고받고, 데이터를 처리할 수 있다.

출처: 위키피디아





핵심 질문

교육용 프로그래밍 언어인 스크래치는 무엇일까?

• 엔트리 다운로드

웹 브라우저 주소창에 <https://play-entry.org>

오브젝트는 캐릭터, 사물, 배경, 글상자로 실행 화면에서 움직일 수 있도록 명령 블록을 통해 작성된다. 블록 꾸러미는 블록, 모양, 소리, 속성의 네 가지 탭으로 이루어져 있다. 블록 조립소는 블록 꾸러미에서 블록을 가져와 조립하는 곳, 다시 말해 프로그래밍을 하는 곳이다.

살펴보기

스크래치 프로그램

1 스크래치(Scratch)란?

MIT 미디어랩이 개발한 공개 소프트웨어로 블록 형태의 명령어를 조합하여 프로그래밍을 할 수 있는 대표적인 교육용 프로그래밍 언어이다. 블록을 조합할 때는 드래그 앤 드롭 방식을 사용한다.

스크래치 프로그래밍은 애니메이션, 게임, 음악 등으로 쉽게 표현할 수 있고 웹을 통해서 다른 사람들과 공유할 수 있다.



2 스크래치 실행

- 1 온라인과 오프라인에서 모두 사용할 수 있다. 온라인에서 사용할 수 있는 웹 버전은 웹 브라우저 주소창에 'https://scratch.mit.edu/'를 입력하여 접속하며, 오프라인 버전은 웹사이트 하단의 [지원]-[오프라인 에디터]에서 설치한다.
- 2 [만들기] 메뉴를 클릭하면 새로운 프로젝트 창이 실행된다.
- 3 [탐색하기] 메뉴로는 각 주제별로 전 세계에서 만들어져 공유된 프로젝트를 확인할 수 있다.

3 스크래치 화면

스크래치에서 꼭 알아 두어야 할 개념은 스프라이트, 무대, 스크립트이다. 이 세 가지 개념이 어우러져 스크래치 프로그램이 완성된다.



114 • Ⅲ부 문제 해결과 프로그래밍

2 자료의 입력과 출력

1 키보드를 통한 입력과 출력

키보드를 이용하여 문자 자료를 입력받은 컴퓨터가 프로그래밍 스크립트에 따라 처리하고, 그 처리 결과를 말풍선으로 출력하는 프로그램을 작성해 보자.

핵심 명령 블록 문자의 입력과 출력

카테고리	명령 블록	기능
이벤트	클릭했을 때	스크래치 프로그램을 시작하는 명령어
형태	Hello! 말하기	입력한 내용을 말풍선으로 나타냄
동작	10 만큼 움직이기	설정한 값만큼 이동
	What's your name? 묻고 기다리기	질문을 하고 입력이 있을 때까지 기다림
	대답	[묻고 기다리기] 명령이 실행되고 키보드로 문자 또는 숫자가 입력되면 저장

따라 하기 고양이 인사하면서 앞으로 움직이는 프로그램

- 1 고양이 스프라이트 선택하기
- 2 이벤트 카테고리에서 '클릭했을 때' 블록 스크립트 영역으로 가져오기
- 3 '형태' 카테고리의 'Hello! 말하기' 블록을 가져와 '안녕하세요?'라고 바꾸기
- 4 '동작' 카테고리의 '10 만큼 움직이기' 가져오기
- 5 '움직였을 때' 클릭하여 실행하기

핵심 명령 블록 방향키 입력과 출력

카테고리	명령 블록	기능
이벤트	스페이스 키를 눌렀을 때	키보드의 특정 키를 눌러 스크립트가 시작
동작	60 도 방향 보기	방향을 나타내는 명령어

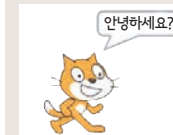
• 스크립트 탭의 10개 카테고리



도전하기

What's your name? 묻고 기다리기  
대답 블록을 이용하여 자기를 소개하는 프로그램을 작성해 보자.

• 실행 결과



도전하기

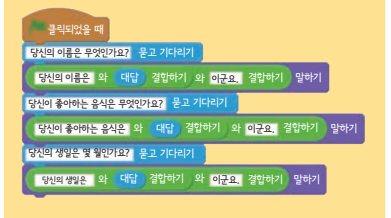
방향키와 A S D W 키를 활용하여 두 마리의 펭귄이 무대에서 움직이도록 프로그램을 작성해 보자.

핵심 질문

스크래치를 이용한 프로그래밍에서 키보드를 통한 자료의 입력과 출력은 어떻게 이루어질까?

- 핵심 명령 블록의 카테고리별로 교과서에 제시된 것 외에 다른 것들도 다양하게 사용해 볼 수 있게 한다.
- 기본적인 프로그램 작성을 마치면 도전하기를 해 보도록 한다(하단 자료 참고).

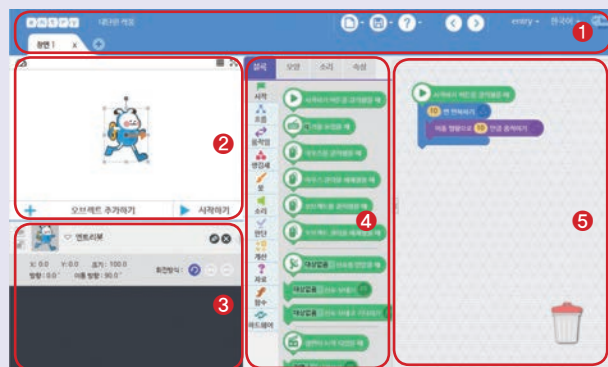
도전하기 예시 답안



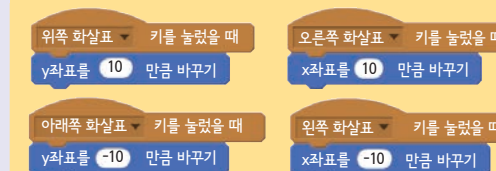
참고자료

• 엔트리(<https://play-entry.org>)

교육용 프로그래밍 언어를 활용한 만들기 화면은 오른쪽 그림과 같이 구성되어 있다. 상단 메뉴(1), 화면의 맨 왼쪽 흰 공간은 오브젝트의 실행 화면(2), 하단은 오브젝트 목록 영역(3), 가운데는 명령어 블록 꾸러미(4), 오른쪽은 블록 조립소 영역(5)이다.



도전하기 예시 답안 ①



도전하기 예시 답안 ②



핵심 질문

스크래치를 이용한 프로그래밍에서 키보드를 통한 자료의 입력과 출력은 어떻게 이루어질까?

- 핵심 명령 블록의 카테고리별로 교과서에 제시된 것 외에 다른 것들도 다양하게 사용해 볼 수 있게 한다.
- 기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다(하단 자료 참고).
- 작품 공유하기를 위해 스크래치의 스튜디오 기능에 대해 소개한다.
- 기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다(하단 자료 참고).

도전하기 예시 답안

이 스프라이트를 클릭했을 때  
pop 재생하기  
다음 모양으로 바꾸기  
10 만큼 움직이기

도전하기  
스프라이트를 마우스로 클릭할 때 소리를 재생하면서 모양이 바뀌고 이동하는 프로그램을 작성해 보자.



따라 하기 고양이 키보드 오른쪽 화살표 키로 이동시키기  
오른쪽 화살표 키를 눌렀을 때  
90도 방향 보기  
10 만큼 움직이기  
1 스페이스 키를 눌렀을 때 블록을 스크립트 창으로 드래그하여  
오른쪽 화살표 선택  
90도 방향 보기 블록 가져오기  
10 만큼 움직이기 블록 가져오기

2 마우스를 통한 입력과 출력  
마우스를 클릭하면 스프라이트가 실행되는 프로그램을 작성해 본다.

카테고리	명령 블록	기능
이벤트	이 스프라이트가 클릭될 때	마우스로 클릭할 때 아래 블록 실행
형태	다음 모양으로 바꾸기	스프라이트의 모양을 모양 탭의 목록 순으로 바꾸기
움직임	무한 반복 위치로 이동하기	스프라이트가 선택한 스프라이트 또는 마우스 포인터의 위치로 이동
펜	펜 내리기	스크립트가 선을 그릴 수 있도록 함

따라 하기 마우스가 클릭하면 모양 바꾸기  
스프라이트 스크립트 실행 결과  
이 스프라이트가 클릭될 때 다음 모양으로 바꾸기

스프라이트	스크립트	실행 결과
	클릭했을 때 무한 반복하기 마우스 포인터 위치로 이동하기	
	클릭했을 때 펜 내리기 무한 반복하기 마우스 포인터 위치로 이동하기	

참고 자료

다양한 그래픽 효과

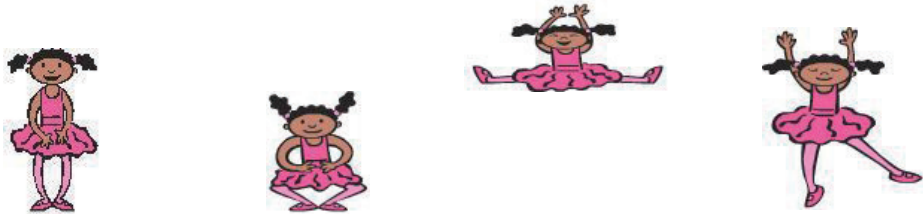
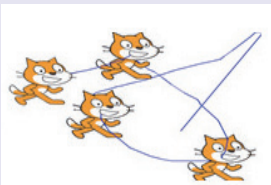
고양이 스프라이트에 다양한 그래픽 효과 적용해 보기(색깔, 어안 렌즈, 소용돌이, 픽셀화, 모자이크, 밝기, 반투명)

색깔	어안 렌즈	소용돌이	픽셀화	모자이크	밝기	반투명

따라 하기 심화 학습

마우스 포인터 위치로 이동하기와 마우스 클릭하기를 동시에 사용해 보기

클릭했을 때  
펜 내리기  
무한 반복하기  
마우스 포인터 위치로 이동하기  
만약 마우스를 클릭했는가? 라면  
도장찍기



비슷한 듯 다른 듯

	모양을 모양2 (으로 바꾸기	다음 모양으로 바꾸기
비슷한 점	스프라이트의 모양을 바꾸어 준다.	
다른 점	저장된 모양 중 선택하여 바꿀 수 있다.	저장된 모양을 순서대로 바꾸어 준다. 마지막 모양 다음에는 다시 처음 모양으로 돌아간다.

	100 만큼 움직이기	x좌표를 100 만큼 바꾸기
비슷한 점	입력한 숫자만큼 스프라이트를 이동시킨다.	
다른 점	스프라이트가 보는 방향으로 이동시킨다. 방향만 바꾸어 주면 상하좌우 모든 방향으로 이동시킬 수 있다.	스프라이트의 방향과 상관없이 스프라이트의 x 좌표를 바꾼다. 스프라이트를 좌우로만 이동시킬 수 있다.

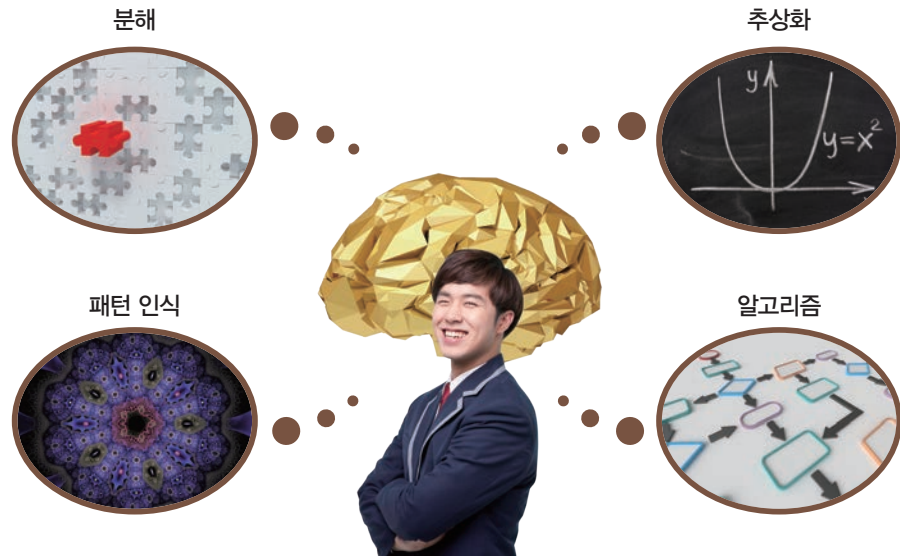
	난수	난수 변수 보이기 난수 변수 숨기기
비슷한 점	무대의 변수나 리스트를 보이게 하거나 숨길 수 있다.	
다른 점	체크 박스를 클릭해야 변수나 리스트를 보이게 하거나 숨길 수 있다. 무대를 전체 화면으로 바꾸었을 때는 체크 박스를 클릭할 수 없다.	스크립트 속에 블록을 넣어서 변수나 리스트를 보이게 하거나 숨길 수 있다. 블록을 사용하기 때문에 무대를 전체 화면으로 바꾸었을 때에도 사용이 가능하다.

	메시지1 방송하기	메시지1 방송하기 기다리기
비슷한 점	다른 블록들이 실행되도록 명령을 내린다.	
다른 점	명령을 내린 뒤에 밑에 있는 블록들을 바로 실행한다.	명령을 내린 뒤에 명령을 받은 블록들이 다 실행될 때까지 기다렸다가 아래 블록을 실행한다.

	방송하기	추가 블록
비슷한 점	사용자가 직접 만들어서 사용한다.	
다른 점	다른 스프라이트에서도 사용할 수 있다. 매개 변수가 없다.	블록을 정의한 스프라이트에서만 사용할 수 있다. 매개 변수를 추가하여 사용할 수 있다.

• 컴퓨팅 사고력의 실습 영역 평가를 위한 수행 평가 문항 예시(Brennan, 2012)

CT 실습 영역	평가 내용
실험과 반복하기	어떻게 프로젝트를 개발했는지 순서대로 이야기해 보자.
	여러분이 프로젝트를 진행하면서 얼마나 다양한 시도를 해 보았는가?
	어떤 수정 사항들이 있었는가? 그리고 왜 수정하게 되었는가?
	여러분이 프로젝트를 만들면서 노력한 점이나 새로운 것들을 시도할 때 사용한 다양한 방법을 이야기해 보자.
테스트와 디버깅하기	프로젝트를 실행한 결과가 여러분이 생각한 것과 달랐을 때 어땠는가?
	문제의 원인을 살펴보기 위해 스크립트를 어떻게 분석했는지 이야기해 보자.
	여러분이 어떻게 바꾸고 수정했는지 이야기해 보자.
	문제를 해결하기 위해 고려한 다른 방법은 무엇이었는지 이야기해 보자.
재사용과 재구성하기	다른 프로젝트를 실행해 보고 스크립트를 살펴 보면서 영감을 받은 것을 이야기해 보자.
	다른 프로젝트의 일부분을 선택하고 여러분의 작품에 활용한 방법을 이야기해 보자.
	기존의 프로젝트를 개선하기 위해 어떻게 수정했는가?
	작업을 하거나 영감을 받은 사람들에게 어떻게 신뢰감을 주었는가?
추상화와 모듈화하기	프로젝트에서 필요한 스프라이트를 결정하고 어디로 보낼지 어떻게 결정했는가?
	프로젝트에 필요한 스크립트를 어떻게 정하고 만들었는가?
	다른 사람들이 이해하기 쉽도록 하기 위해 어떻게 스크립트를 만들었는가?



### 3-3 변수와 연산

주변 기온에 따라 온도계의 수치가 변하고 달리는 자동차의 속도계도 상황에 따라 변한다. 프로그래밍에서는 수치로 변하는 수치를 어떻게 처리하는지 알아보자. 그리고 프로그램에서는 입력된 자료와 수치를 어떤 방법으로 연산을 하는지도 살펴보자.

#### 1 변수

##### 1 변수의 이해

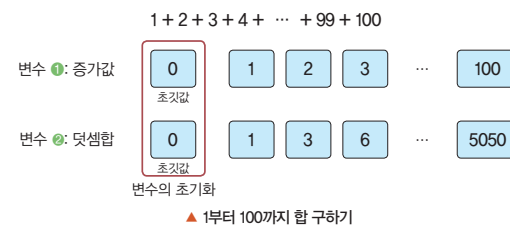
입력된 자료에 따라 변하는 값을 변수라 한다. 하나의 상자에 다양한 물건을 담을 수 있는 것처럼 프로그래밍에서도 변하는 값을 담는 공간을 변수라고 한다. 변수의 유형에는 정수나 실수와 같은 숫자가 저장되는 숫자형 변수, 문자를 저장하는 문자형 변수, 참과 거짓의 자료가 저장되는 논리형 변수가 있다.



##### 2 변수 만들기

1부터 100까지 합을 구하는 프로그램을 구해 보자.

1 + 2 + 3 + ... + 100까지 1씩 증가되는 값과 덧셈 값을 보관하는 변수가 필요하다. 변수를 선언하고 처음으로 값을 저장하는 것을 '변수의 초기화'라고 한다.



- **상수**  
변수와는 달리 고정된 값을 나타낸다.
- 변수는 자유롭게 이름을 붙일 수 있으나, 가능하면 결과 값의 의미를 명확히 알 수 있는 변수명으로 정하고, 초깃값을 부여한다.
- **변수의 초기화**  
클릭했을 때  
증가값 올(를) 0 로 정하기

#### 변수 만들기

1 데이터 카테고리의 변수 만들기

스크립트 모양 소리

동작 형태 소리 펜 데이터

이벤트 제어 관찰 연산 추가블록

변수 만들기

변수 이름 만들기

새로운 변수

변수 이름 정수

종료 스크립트(이벤트) 사용 ○ 이 스크립트에(이벤트) 사용

확인 취소

#### • 비슷한 듯 다른 듯 •

	펜 색깔을 (으)로 정하기	펜 색깔을 0 (으)로 정하기
비슷한 점	• 스프라이트의 펜 색깔을 정해 준다.	
다른 점	• 화면에 있는 색깔을 뽑아서 정한다. • 색깔을 정해 주는 칸을 클릭한 후 컴퓨터 화면에서 원하는 색을 클릭하면 정할 수 있다.	• 숫자를 이용해서 펜 색깔을 정한다. • 숫자 0: 빨강, 숫자 70: 초록, 숫자 130: 파랑

#### 핵심질문

변수의 개념을 이해하고 변수와 연산자를 활용한 프로그램을 작성할 수 있는가?

#### 핵심질문

변수는 어떻게 만들 수 있을까? 명령 블록이 실행될 때 변수의 값이 어떻게 변할까?

- 변수의 기능과 역할에 대해 소개한다. 변수와 변수 값의 의미를 이해할 수 있도록 지도한다.
- 기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해보도록 한다(하단 자료 참고).

#### 변수 만들기 심화 학습

##### 1~100까지의 합 구하기



클릭했을 때

증가값 올(를) 0 로 정하기

덧셈할 올(를) 0 로 정하기

100 번 반복하기

증가값 올(를) 1 만큼 바꾸기

덧셈할 올(를) 증가값 만큼 바꾸기

덧셈할 말하기

0.1 초 기다리기



## 함께 해 보기 예시 답안

- ~만큼 바꾸기: 처음 값에서 설정한 만큼 더해져 바꾸어 주어야 한다. 처음 값과 결과 값이 다르다.
- ~로 정하기: 처음 값을 설정한 값으로 무조건 바꾸어 준다.

## 도전하기 예시 답안

클릭했을 때  
점수 올(름) 70 로 정하기  
점수 말하기  
1 초 기다리기  
점수 올(름) 20 만큼 바꾸기  
점수 말하기  
1 초 기다리기  
점수 올(름) -40 만큼 바꾸기  
점수 말하기

## 도전하기 예시 답안

20초가 지나면 야옹 소리가 나는 타이머 만들기

클릭했을 때  
시간 올(름) 0 로 정하기  
무한 반복하기  
시간 올(름) 1 만큼 바꾸기  
3 초 기다리기  
만약 시간 = 20 라면  
시간 올(름) 0 로 정하기  
meow 재생하기  
1 초 기다리기  
모두 멈추기

### 함께 해 보기

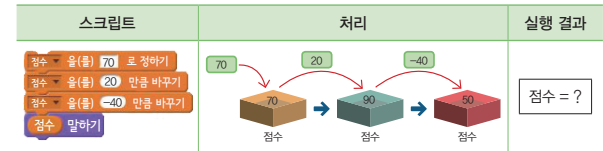
#### 변수 명령 블록 비교

점수 올(름) 10 로 정하기 와  
점수 올(름) 10 만큼 바꾸기 의  
차이점이 무엇인지 알아 보자.

### 도전하기

'점수' 변수 값이 변화하는 과정을 실행 창에서 확인할 수 있도록 블록을 추가해 보자.

'점수' 변수를 만든 후 다음과 같이 세 개의 명령 블록이 실행되면 '점수' 변수의 값은 어떻게 변할까? 다음 '점수' 변수에 최종적으로 저장되는 값은 얼마일까?



### 핵심 명령 블록 움직이는 고양이

카테고리	명령 블록	기능
이벤트	이 스프라이트가 클릭될 때	스프라이트를 클릭할 때 아래 블록 실행
연산	1 부터 10 사이의 난수	입력한 두 수 사이에서 선택된 무작위 수의 값
동작	x: 0 y: 0 로 이동하기	스프라이트를 입력한 X좌표, Y좌표로 이동
형태	색깔 효과를 25 만큼 바꾸기	스프라이트에 선택한 효과를 입력한 값만큼 주고 색깔, 밝기, 투명도 등을 선택
데이터	고양이 변수 보이기	추가된 변수가 실행 화면에 나타나도록 함

따라 하기 스프라이트를 클릭할 때마다 모양을 바꾸면서 X좌표 위치를 변수로 입력받고 이동하기

이 스프라이트가 클릭될 때  
고양이 X좌표 올(름) -200 부터 200 사이의 난수 로 정하기  
다음 모양으로 바꾸기  
x: 고양이 X좌표 y: 0 로 이동하기  
고양이 X좌표 올(름) 2 초동안 말하기



### 작품 공유하기

스크래치 홈페이지에 있는 프로젝트를 살펴보거나 특별한 타입의 프로젝트를 찾기 위해 [탐색하기] 메뉴를 클릭해 보자.

118 • 11부 문제 해결과 프로그래밍



### 스프라이트를 클릭할 때마다 점수 1씩 증가시키기

사과, 오렌지, 바나나 스프라이트를 클릭할 때마다 점수를 1씩 증가시키고, 결과 확인 버튼을 클릭하면 결과 점수를 보여 주도록 하자.

스프라이트	스크립트	실행 결과
	클릭했을 때 이 스프라이트가 클릭될 때 사과 올(름) 0 로 정하기 사과 올(름) 1 만큼 바꾸기 사과 계수는 소리 재생하기 이 스프라이트가 클릭될 때 사과 변수 보이기	사과 0 오렌지 0 바나나 1 결과 확인 스프라이트

## 2 연산자

### 1 연산자의 종류

연산자는 프로그래밍 과정에서 계산 처리를 위해 사용되는 중요한 요소이다. 산술, 논리, 비교 연산자를 사용한다. 산술 연산자는 사칙 연산을 처리할 때, 논리 연산자는 참인지 거짓인지의 논리적인 판단을 할 때, 비교 연산자는 크기를 비교할 때 사용한다.

### 핵심 명령 블록 연산 블록

구분	카테고리	명령 블록	기능
연산	산술 연산	덧셈, 뺄셈, 곱셈, 나눗셈	덧셈, 뺄셈, 곱셈, 나눗셈
	비교 연산	~ 크다, ~ 같다, ~ 작다	~ 크다, ~ 같다, ~ 작다
논리 연산	그리고	양쪽 조건이 맞을 때의 논리 값	양쪽 조건이 맞을 때의 논리 값
	또는	두 조건 중 하나 이상 맞을 때의 논리 값	두 조건 중 하나 이상 맞을 때의 논리 값
	가(이) 아니다	조건이 거짓일 때 참이 되는 논리 값	조건이 거짓일 때 참이 되는 논리 값

• 프로그래밍 언어에 따라 연산자를 표시하는 방법이나 사용법이 조금씩 다르다.

### • 논리곱

A	그리고	B	→	C
참		참		참
참		거짓		거짓
거짓		참		거짓
거짓		거짓		거짓

### • 논리합

A	또는	B	→	C
참		참		참
참		거짓		참
거짓		참		참
거짓		거짓		거짓

### • 부정

A	~A
참	거짓
거짓	참

119 • 11부 문제 해결과 프로그래밍

## 함께 해 보기 예시 답안

클릭했을 때  
사과 올(름) 0 로 정하기  
사과 변수 숨기기

이 스프라이트를 클릭했을 때  
사과 올(름) 1 만큼 바꾸기

사과 스프라이트  
(다른 과일도 동일)

이 스프라이트를 클릭했을 때  
사과 변수 보이기 결과 확인 스프라이트

## 핵심 질문

연산자란 무엇일까? 연산자의 종류에는 무엇이 있고 연산자를 어떻게 활용할까?

• 기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다(하단 자료 참고).

## 참고자료

### • 추가 선택 활동 •

▶ **리스트:** 변수들을 순서로 세워놓은 것을 가지고 한줄로 세운 것. 여러 자료를 추가하거나 삭제하며 사용할 수 있음.

• **리스트의 활용 심화 학습**

▶ 설문 조사 질문지 만들기

클릭했을 때  
모두 번째 항목을 자기 소개 에서 삭제하기  
이름이 무엇인가요? 묻고 기다리기  
대답 항목을 자기 소개 에 추가하기  
생일은 언제 인가요? 묻고 기다리기  
대답 항목을 자기 소개 에 추가하기  
좋아하는 음식이 무엇인가요? 묻고 기다리기  
대답 항목을 자기 소개 에 추가하기  
취미는 무엇인가요? 묻고 기다리기  
대답 항목을 자기 소개 에 추가하기

### • 추가 선택 활동 •

▶ **연산자 우선순위:** 다음 계산 결과 값을 비교하여 보자.

$$3 + 5 * 2 - 7$$

$$(3 + 5) * (2 - 7)$$

$$((3 + 5) * 2) - 7$$

a 키를 눌렀을 때  
계산결과 올(름) 3 + 5 \* 2 - 7 로 정하기  
계산결과 말하기  
b 키를 눌렀을 때  
계산결과 올(름) 3 + 5 \* 2 - 7 로 정하기  
계산결과 말하기  
c 키를 눌렀을 때  
계산결과 올(름) 3 + 5 \* 2 - 7 로 정하기  
계산결과 말하기

**함께 해 보기 예시 답안**

큰 수, 작은 수

클릭했을 때

1~100 사이의 숫자 묻고 기다리기

만약 **대답 < 50**

50보다 작은 수입니다. 말하기

아니면

50보다 같거나 큰 수입니다. 말하기

**함께 해 보기 심화 학습 예시**

클릭했을 때

1~100 사이의 숫자 묻고 기다리기

만약 **대답 나누기 2 의 나머지 = 0** 라면

짝수입니다. 말하기

아니면

홀수입니다. 말하기

## 참/고/자/료

### 따라 하기 심화 학습

- ▶ x 좌표, y 좌표로 말하는 고양이 만들기

클릭했을 때

x: 0 y: 0 로 이동하기

무한 반복하기

x좌표 와 y좌표 결합하기 말하기

x좌표를 -100 부터 100 사이의 난수 만큼 바꾸기

y좌표를 -100 부터 100 사이의 난수 만큼 바꾸기

### 2 연산자의 활용

계산 블록을 통해 숫자 값과 논리 값을 이용하고, 스프라이트의 움직임을 정하거나 조건을 줄 때 사용한다.

#### 따라 하기 총점과 평균 구하기

클릭했을 때

정보 올(을) 0 로 정하기

과학 올(을) 0 로 정하기

기술 올(을) 0 로 정하기

총점 올(을) 0 로 정하기

평균 올(을) 0 로 정하기

정보 점수를 묻고 기다리기

정보 올(을) 대답 로 정하기

과학 점수를 묻고 기다리기

과학 올(을) 대답 로 정하기

기술 점수를 묻고 기다리기

기술 올(을) 대답 로 정하기

총점 올(을) 정보 과학 기술 로 정하기

평균 올(을) 총점 / 3 로 정하기

총점과 평균 결합하기 말하기

평균과 평균 결합하기 말하기

- 1 '정보', '과학', '기술', '총점', '평균' 변수를 만들어 0을 초기값으로 정한다.
- 2 What's your name? 묻고 기다리기 과 대답 을 이용하여 정보, 과학, 기술 점수를 입력받는다.
- 3 블록 과 을 이용하여 총점과 평균을 구하는 식으로 만들어 총점과 평균 변수에 입력한다.
- 4 Hello! 말하기 블록과 hello 와 world! 결합하기 블록을 이용하여 총점과 평균을 출력하도록 한다.



큰 수, 작은 수

1에서 100 사이 임의의 숫자를 입력받은 뒤에 입력받은 수가 50보다 작으면 "작은 수입니다", 50과 같거나 크면 "같거나 큰수입니다"를 출력해 보자.

명령 블록	실행 결과
1~100 사이의 숫자 묻고 기다리기	1~100 사이의 숫자
50보다 작은 수입니다. 말하기	
만약 라면 대답	
아니면	
50보다 작은 수입니다.	

프로그램 심화시키기

50보다 작거나 큰 수를 비교하는 스크립트를 수정하여 짝수와 홀수를 판단하는 프로그램을 작성해 보자.



10개의 블록

10개의 명령 블록으로 어떤 프로그램을 만들 수 있을까? 다음 명령 블록을 이용하여 프로젝트를 만들어 보자. 각 블록을 복사하여 한 번 이상 사용해도 된다. 단, 각각의 블록을 적어도 한 번은 꼭 사용해야 한다.

아용 끝까지 재생하기 10 번 반복하기

x: 0 y: 0 로 이동하기

1 초 동안 x: 0 y: 0 으로 이동하기

숨기기 크기를 100 %로 정하기

보이기 Hello! 올(을) 2 초 동안 말하기

이 스프라이트를 클릭했을 때 1 초 기다리기

#### 1 각각의 블록을 적용하면서 자신의 아이디어를 표현해 보자.

- 다른 스프라이트, 모양, 배경으로 꾸며 보자.
- 10개의 블록으로 만들 수 있는 프로젝트가 얼마나 많은지 알아보자.
- 친구들과 아이디어를 브레인스토밍하여 각각의 창작물을 공유해 보자.

#### 2 평가하기 다음 물음에 대한 여러분의 생각을 빈칸에 써 보자.

- 명령어 전체를 사용할 때보다 10개의 블록만 사용할 때 어떤 점이 쉬웠는가?

- 명령어 전체를 사용할 때보다 10개의 블록만 사용할 때 어떤 점이 어려웠는가?

- 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

#### 3 심화하기 변수, 소리, 형태, 연산 블록 중 10개를 더 추가하여 나만의 작품을 만들어 보자.

### • C-B-T 전략(제약 중심 모형)

다음과 같은 여러 가지 제약(Constraint) 조건을 제시하여 주어진 조건하에 만들어 보고(Build) 문제를 해결해 봄으로써(Test) 각 명령어의 특성을 익히고 문제 해결력을 기를 수 있다.

- 1 주어진 10개의 블록을 한 번씩만 사용하도록 한다.
- 2 주어진 10개의 블록을 여러 번 중복해서 사용하도록 한다.
- 3 여러 개의 블록 중에서 10개만 골라서 사용하도록 한다.

**함께 해 보기 예시 답안**

1

Hello!

Hello!

- 평가지 외에도 다양한 방법을 통해 평가가 이루어질 수 있게 한다.

- 테스트와 디버깅이 자유롭게 이루어질 수 있는 분위기를 조성한다.

### • 비슷한 듯 다른 듯 •

	음량	음량
비슷한 점	• 소리를 숫자 값으로 바꾸어 나타낸다.	
다른 점	• 스프라이트, 즉 컴퓨터가 밖으로 내는 (out-put) 소리의 크기를 숫자 값으로 나타낸다.	• 외부에서 마이크를 통해 컴퓨터로 들어가는 (in-put) 소리의 크기를 숫자 값으로 나타낸다.

### • 10블록 예시

이 스프라이트를 클릭했을 때

x: 0 y: 0 로 이동하기

크기를 100 %로 정하기

보이기

10 번 반복하기

아용 끝까지 재생하기

Hello! 올(을) 2 초 동안 말하기

크기를 10 만큼 바꾸기

10 만큼 움직이기

### • 디버깅 소스 자료 제시

<http://scratch.mit.edu/projects/67634530>  
<http://scratch.mit.edu/projects/67636752>  
<http://scratch.mit.edu/projects/67638958>  
<http://scratch.mit.edu/projects/67639058>  
<http://scratch.mit.edu/projects/67639810>

# 제어 구조

## 핵·심·질·문

순차, 선택, 반복의 개념과 원리를 이해하고 세 가지 구조를 활용한 프로그램을 작성할 수 있는가?

### D-M-M 전략(시연 중심 모형)

시연-모방-제작의 단계를 거치며 교사가 모델이 되는 학습 활동의 시연을 거쳐 학습자들이 질문과 대답을 통한 모방 그리고 반복적으로 단계적, 독립적 연습을 통하여 제작하는 활동 중심 모형을 적용해 보자.

시연 (Demonstration)	설명하기, 시범 보이기, 예시 보이기
모방 (Modeling)	따라하기, 질문과 답변
제작 (Making)	만들기, 반복 활동으로 기능 습득

## 참/고/자/료

### 순차·선택·반복 구조 비교

- 1 순차(concatenation):** 순차는 구문 순서에 따라서 순서대로 수행된다는 것이다.
- 2 반복(repetition):** 반복은 프로그램이 특정 상태에 도달할 때까지 구문을 반복하여 수행하거나, 집합체 각각의 원소들에 대해 어떤 구문을 반복 수행하는 것이다. 보통 while, repeat, for, do..until 같은 키워드로 표현한다.

		
회전방식을 회전하기로 정하기	회전방식을 왼쪽-오른쪽으로 정하기	회전방식을 회전하지 않기로 정하기
회전 방식: 	회전 방식: 	회전 방식: 
회전하기	왼쪽 - 오른쪽	회전하지 않기



## 3-4 제어 구조

우리는 일상생활에서 일을 할 때 순서대로 처리하거나 반복 또는 조건에 따라 선택하여 처리한다. 주어진 문제를 해결하기 위해 프로그램으로 구현할 때도 같은 내용을 여러 번 반복하거나 조건에 따라 일련의 실행 순서가 정해지는 절차 구조로 처리한다. 이 단원에서는 순차, 반복, 선택 구조를 이해하고, 효율적인 프로그램 작성을 위해 어떤 명령을 사용하는지 살펴보자.

### 1 순차 구조

종이접기를 할 때는 도안을 보고 순서대로 접는다. 이처럼 주어진 문제를 해결하기 위해 처음부터 끝까지 정해진 순서에 따라 차례대로 명령이 나열된 것을 순차 구조라 한다.  
프로그래밍 언어로 정사각형을 어떻게 그릴 수 있을까? 그림을 그리기 위해 먼저 펜 카테고리 명령 블록을 살펴보자.

#### 핵심 명령 블록 그림 그리기 블록

카테고리	명령 블록	기능 설명
펜	펜 내리기	스크립트가 선을 그리도록 함
	선 색깔을 (의)로 정하기	선의 색깔을 바꿈
	선 굵기를 1 만큼 바꾸기	선의 굵기를 바꿈
	채우기	그린 선을 모두 지움

#### 따라 하기 정사각형 그리기

- 1 펜 카테고리 펜 내리기 블록을 드래그하여 스크립트 영역에 둔다.
- 2 100 만큼 움직이기 120 도 돌기 명령어를 입력한다.
- 3 한 번 더 100 만큼 움직이기 120 도 돌기 명령어를 입력한다.
- 4 또 한 번 더 100 만큼 움직이기 120 도 돌기 명령어를 입력한다.

**프로그램 개선**  
정사각형을 그리는 프로그램을 작성할 때 어떤 점이 불편했는가? 그 개선 방법을 토론해 보자.

### 2 반복 구조

프로그램을 작성할 때 중복되는 부분을 효율적으로 처리하기 위해 반복 구조가 필요하다. 반복 구조는 조건이 만족하는 동안 정해진 구간을 반복 명령어를 이용하여 수행한다. 반복을 실행하기 위한 '반복 횟수'와 조건이 만족했을 때 실행할 '실행 영역'으로 구분된다.

#### 핵심 명령 블록 반복 명령 블록

구분	명령 블록	기능
제어	10 번 반복하기	블록 안의 스크립트를 정해진 횟수만큼 반복
	무한 반복하기	블록 안의 스크립트를 무한 반복
	까지 반복하기	조건이 참이 될 때까지 실행 영역 안의 스크립트를 반복 실행

#### 따라 하기 삼각형 그리기

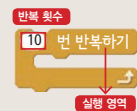
- 1 펜 카테고리 펜 내리기 블록을 드래그하여 스크립트 영역에 둔다.
- 2 제어 카테고리의 10 번 반복하기 블록을 드래그한 후
- 3 3 키를 눌렀을 때 펜 내리기 100 만큼 움직이기 120 도 돌기 1 초 기다리기 1 초 기다리기 펜 올리기 를 입력한다.
- 4 100 만큼 움직이기 반시계 방향으로 120도 회전하도록 360 / 3 도 돌기 연산을 입력한다.
- 5 펜 올리기 블록을 드래그한다.
- 6 삼각형을 그리는 모습을 차근차근 보여 주기 위해 움직이기와 돌기 사이에 1 초 기다리기 를 넣는다

### 3 선택 구조

조건에 따른 참과 거짓 여부에 따라 다른 처리를 하기 위해 흐름을 변경해야 하는 것을 선택 구조라 한다. 조건 명령의 조건식에 따라 다음에 실행할 명령을 선택하여 실행한다.

#### 따라 하기 심화 학습

▶ 무한 반복을 이용하여 가운데가 뚫린 원 그리기



각에 의한 꼭짓점을 표현하기 위한 '돌기' 값을 추상화하면 '360도/선의 수' 또는 '180 - 내각의 크기'가 된다.

#### 도전하기

정사각형 그리기와 삼각형 그리기 스크립트를 이용하여 다양한 도형을 그려 보자.

정다각형을 그리기 위한 돌기 값을 정하기 위해 여러 정다각형의 특징을 조사하고 패턴을 찾아보자.

#### 정다각형의 핵심 요소

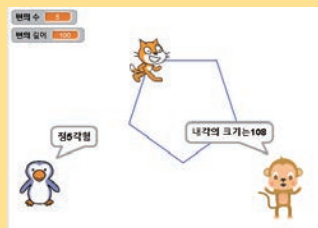
정삼각형, 정사각형, 정오각형, 정육각형 등 정다각형의 공통점은 무엇일까? 이러한 도형들을 그리기 위한 핵심 요소는 선분(움직이기)과 각에 의한 꼭짓점(돌기)이다.

• 기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다(하단 자료 참고).

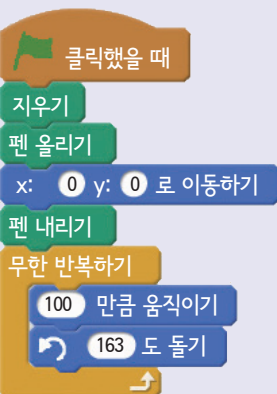
#### 도전하기 예시 답안

##### 자동 정다각형 그리기

- 클릭했을 때 지우기 펜 내리기 90 도 방향 보기
- 클릭했을 때 정몇각형을 그릴까요?(숫자만) 묻고 기다리기 변의 수 을(를) 대답 로 정하기 크기는 얼마로 할까요?(10~200) 묻고 기다리기 변의 길이 을(를) 대답 로 정하기 변의 수 번 반복하기 변의 길이 만큼 움직이기 360 / 변의 수 도 돌기 그러라 방송하기



- 그러라 을(를) 받았을 때 정 와 변의 수 결합하기 와 값형 결합하기 말하기
- 그러라 을(를) 받았을 때 내각의 크기를 와 180 - 변의 수 결합하기 말하기



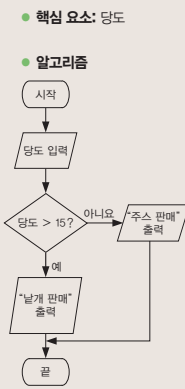


**도전하기 예시 답안**

```

클릭했을 때
난수 = 올(를) 1 부터 4 사이의 난수 로 정하기
만약 난수 = 1 라면
  x좌표를 10 만큼 바꾸기
아니면
  만약 난수 = 2 라면
    x좌표를 -10 만큼 바꾸기
  아니면
    만약 난수 = 3 라면
      y좌표를 -10 만큼 바꾸기
    아니면
      y좌표를 10 만큼 바꾸기
  
```

- 기본적인 프로그램 작성을 마치면 다양한 심화 학습을 해 보도록 한다(하단 자료 참고).
- 지도서 부록인 스크래치 카드를 이용하여 모둠별 스크래치 카드 놀이를 해 보자. 스크래치의 기본 기능과 다양한 기본 모듈을 익히기에 유용하다.
- 학습자 수준과 여유 시간을 고려하여 step 1~3의 내용을 선택하여 활동할 수 있다.



**도전하기**

1~4 사이에 난수 값을 받아 10이면 '동쪽', 20이면 '서쪽', 30이면 '남쪽', 40이면 '북쪽'으로 스프라이트를 이동시키는 프로그램을 만들어 보자.

카테고리	명령 블록	활용
만약	라면	'만약 ~라면' 조건이 맞을 경우 블록 안의 스크립트를 실행
제어	만약 ~라면 아니면	'만약 ~라면' 조건이 맞을 경우 첫 번째 빈 칸 안의 스크립트를 실행, 조건이 맞지 않으면 두 번째 빈 칸 안의 스크립트를 실행

**따라 하기 사과 분류하기**

사과의 당도를 입력받아 15브릭스를 초과하면 '날개 판매', 15브릭스 이하면 '주스 판매'로 구분하도록 출력해 보자.

클릭했을 때

사과의 당도가 얼마입니까? 묻고 기다리기

만약 대답 > 15 라면

날개 판매 말하기

아니면

주스 판매 말하기

- 1 사과 당도를 입력받기 위해 묻고 기다린다.
- 2 제어에서 조건 명령 블록을 선택한다.
- 3 조건식에서 사용될 비교 연산자를 선택한다.
- 4 비교 연산자 >에 대답 변수를 넣고 15와 비교한다.
- 5 조건 명령에 말하기 명령을 추가한다.

**스크래치 카드 놀이**

4명이 한 모둠을 이루어 스크래치 카드를 이용하여 함께 활동해 보자.

**1 Step** 선택한 '스크래치 카드'의 스크립트를 모듬원 모두가 만들면 카드 획득

**2 Step** 획득한 여러 장의 '스크래치 카드'를 이용하여 새로운 작품 만들기

**3 Step** 다른 모듬과 카드를 교환하여 작품을 추가하거나 수정하기

**3-5 프로그래밍 응용**

선봉기, 로봇 청소기, 자동판매기와 같은 생활 속 제품에는 프로그램이 포함되어 있다. 실생활의 다양한 문제 해결을 위해 프로그램을 구현해 보고, 효율적인 프로그램을 완성할 수 있도록 개발 과정을 공유·비교·분석하는 활동을 통해 협력적 프로젝트를 수행해 보자.

**프로젝트 1**

**'사과 받기' 게임 재구성**

'사과 받기' 게임을 실행해 보자.

**'사과 받기' 게임을 해 보고 게임의 내용을 적어 보자.**

'사과 받기' 게임은 바구니를 화살표 키(← →)를 이용하여 좌우로 움직여 가며 떨어지는 사과를 받아 내는 게임이다. 사과를 받을 때 마다 1점씩 올라가며 사과가 바구니에 닿는 순간 소리도 함께 나도록 하였다.

**놀이하기** '사과 받기' 게임을 해 보고 어떤 기능이 있는지 관찰하여 적어 보자.

사과	위에서 아래로 떨어진다. X좌표 값은 무작위로, Y좌표 값은 일정하게 하여 무작위의 위치에서 떨어진다.
바구니	
변수	
음향 효과	사과가 바구니에 닿을 때 '뽕' 소리 내기

**25 차시**

**소단원 지도 GUIDE**

**3-5 프로그래밍 응용**

**프로젝트 1**

**사과 받기 게임**

**핵·심·질·문**

실생활 문제 해결을 위한 소프트웨어를 협력하여 설계, 개발, 비교·분석할 수 있는가?

- **U-M-C 전략(재구성 중심 모형)**  
주어진 프로젝트를 조작해 보는 놀이를 통해 배우고자 하는 학습 모듈을 학생들이 탐색하고 모듈의 수정 과정을 통하여 기능과 개념을 이해한다. 놀이 활동과 수정 활동이 연계된 일련의 재구성 활동을 통해 학생들이 컴퓨팅 사고의 전반적인 능력을 이해하게 한다.

놀이 (Using)	조작, 체험, 놀이, 활용, 탐색
수정 (Modify)	추가 설계, 수정, 확장, 보완
재구성 (reCreate)	재구성, 구현, 개발, 산출

- **재구성 결과 예시**
  - 구성 요소 바꾸기: 사과를 물고기로, 바구니를 꽃게로 바꾸어서 꽃게가 물고기를 잡는 상황 연출. 무대 배경을 바다로 바꾸기
  - 기능 바꾸기
    - 1 명령어의 수정: 떨어지는 물고기의 속도 바꾸기, 꽃게의 방향 조절을 마우스로 따라다니도록 설정, 매개 변수에서 쌓이는 점수 바꾸기, 음향 효과를 '으악'이라는 내 목소리를 녹음해서 바꾸기 등
    - 2 명령어의 복제: 떨어지는 물고기 수 늘리기 등
    - 3 명령어의 추가: 생명 변수, 타이머 변수 넣기. 해당 점수가 되면 장면을 추가하여 다음 장면으로 이동하며 레벨업 나타내기, 번개를 떨어지게 하기 등

- **작품 예시**
  - 움직이는 꽃게 만들기

클릭했을 때

모양을 모양1으로 바꾸기

무한 반복하기

색깔 효과를 25 만큼 바꾸기

만약 마우스 포인터에 닿았는가? 라면

Hello! 말하기

10 번 반복하기

10 만큼 움직이기

다음 모양으로 바꾸기

0.2 초 기다리기

-10 만큼 움직이기

다음 모양으로 바꾸기

0.2 초 기다리기

아니면

please touch me! 말하기

- 놀이하기 답안**
- 바구니: 좌우로 움직인다.
  - 변수: 바구니가 사과에 닿으면 점수 변수에 점수가 추가된다.



• 짝 프로그래밍 기법(pair-programming)  
두 사람이 짝이 되어 한 사람은 코딩을 하고 나머지 한 사람은 코딩 순서를 알려주는 역할로 나누어 프로그래밍하며 일정 시간이 지나면 서로 역할을 바꾸어 교대로 프로그래밍하며 협력하는 기법이다. 이러한 짝 프로그래밍은 다음의 여러 가지 전략을 고려할 수 있다.



- ① 각자의 역할을 분명히 정한다.
- ② 1인당 코딩 시간을 제한한다.
- ③ 1인당 코딩 블록 수를 제한한다.
- ④ 상대방의 코딩 블록을 수정하게 하거나 못하게 한다.

• 이야기를 만들기 위해 설계가 선행되어야 한다. 학생들이 쉽게 이해할 수 있도록 장면을 선정하는 것부터 시작해 본다.

• 장면에는 배경, 캐릭터, 간단히 주고받는 대화가 표현되어야 한다.

프로젝트 2

친구와 이야기 속으로



작은 별에 사는 어린 왕자는 나무를 뽑고 화산을 청소하며 하루하루를 보내고 있었다. 어느 날 피어난 장미를 사랑하게 된 어린 왕자는 장미가 원하는 것은 무엇이든 들어주었다. 하지만 변덕스러운 장미 때문에 어린 왕자는 별을 떠나기로 결심하였다. 여러 별을 여행하던 어린 왕자는 많은 사람을 만나며 다양한 경험을 한다. 지구에 도착한 어린 왕자는 불시착한 조종사를 만나며 이야기를 나누게 된다.

준비하기 짝과 함께 어린 왕자의 이야기 중에서 구현하고 싶은 장면을 선택하여 간단히 그려 보자.

스토리 :

장면 1	장면 2
장면 3	장면 4

수정하기

명령어 수정

- ① 사과가 떨어지는 속도를 변경해 보자.
- ② 바구니의 방향 조절을 키보드 대신 마우스를 따라다니게 해 보자.
- ③ 매개 변수에서 쌓이는 점수를 바꾸어 보자.
- ④ 음향 효과를 바꾸어 보자.
- ⑤ ( )

기능을 바꾸어 볼까?



명령어 복제

- ① 떨어지는 사과의 수를 늘린다.
- ② ( )

명령어 추가

- ① 타이머 변수를 넣어 시간 제한을 주자.
- ② 해당 점수가 되면 다음 단계의 장면으로 이동해 보자.
- ③ ( )

스프라이트를 추가해 볼까?



새로운 스프라이트 ( )의 기능

- ①
- ②
- ③

구현하기

위의 수정 계획을 바탕으로 사과 받기 게임을 완성해 보자.

되돌아보기

- 재구성한 게임을 실행시키고 오류가 있는 부분을 수정해 보자.
- 똑같은 프로젝트를 어떻게 다양하게 변화시켰는지 서로 발표하며 공유해 보자.

• 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

• 소프트웨어 프로젝트 평가 방법

- ▶ 프로젝트 산출 평가: 학기 말 과제, 팀 프로젝트, 산출물 과제 대회 등을 통해 산출된 결과물(코드)을 다면 평가 예 자동 평가 : 명령어 컬러 팔레트 분석, Dr.Scratch 분석/수동 평가: 평가 루브릭을 이용하여 교사가 직접 분석
- ▶ 포트폴리오 분석: 프로젝트 진행 과정에서 나타난 모든 산출물(계획서, 설계 자료, 중간 산출물, 최종 산출물, 참고 자료 등)을 종합적으로 평가
- ▶ 디자인 저널 평가: 프로젝트를 진행하기 위해서 학생들이 구상, 설계, 개발, 디버깅, 성찰한 것 등의 전체적인 과정을 기록한 디자인 저널을 평가하는 방법

- ▶ 인터뷰 평가: 산출 과정에 대하여 학생 개인 또는 집단을 대상으로 창작, 소감, 태도 등을 인터뷰하여 평가하는 방법
- ▶ 기타: 체크리스트 또는 평가표를 이용하여 자기 평가/동료 평가를 활용

• 짝 프로그래밍

2명 이상의 학습자가 기능적, 인지적 상호 작용을 하면서 의사소통과 협력을 향상시키고 전문성을 신장시키기 위해 효과적인 프로젝트 수업으로 발전시킬 수 있다. 제약 조건은 학생들의 논리적 사고를 더욱 날카롭게 할 뿐만 아니라 제약 조건의 범위 안에서 창의적 사고를 이끌어 낸다.

- 짝 프로그래밍의 구성: 학생과 학생(2명, 3명), 교사(전문가)와 학생, 학생 개인과 학생 집단, 학생 집단과 학생 집단, 한 집단 프로그래밍(몸 프로그래밍 또는 팀 프로그래밍)

• 짝 프로그래밍의 제약 조건

- ▶ 제한된 블록의 개수(1개, 5개 등)를 서로 교차하여 개발
- ▶ 서로 말을 하지 말고 블록만 하나씩 놓으면서 상대의 의도를 파악하게 하는 경우
- ▶ 짝이 내가 만든 블록을 수정 가능하게 하는 경우와 수정하지 못하게 막아 놓는 경우
- ▶ 스크래치 블록(명령어군)을 구분하여 자신이 맡은 색의 블록만 사용
- ▶ 시간을 두고 역할 바꾸기(30초, 1분 등)
- ▶ 미완성된 프로그램을 수정하거나 디버깅하기

출처: 『컴퓨팅 사고력을 위한 소프트웨어 교육』, 생능출판사

• 각 장면마다 움직이는 2개의 캐릭터를 만들고, 추가 블록 만들기 연습도 함께 해 보면서 자동화의 효율성을 이해할 수 있도록 한다.

• 멀티미디어 자료로는 자신의 얼굴 사진을 찍어 활용하기, 배경 음악 넣기, 자신의 음성을 녹음하여 넣기 등으로 활용할 수 있다.

• 대화를 구성할 때는

메시지1 - 발송하기

메시지1 ▾ 울(를) 받았을 때

메시지1 ▾ 발송하고 기다리기

블록을 활용하도록 하여 대화 조절 방법을 익히고, 이야기의 논리적인 흐름을 구성할 수 있도록 한다.

수정하기

1 사용된 스프라이트의 주요 기능을 설계해 보자.

스프라이트	주요 기능(알고리즘)

2 각 장면에 사용되는 여러 가지 기능과 요소를 적어 보자.

장면	1	2	3	4
캐릭터				
애니메이션 멀티미디어				
스프라이트 간의 상호작용				

3 어린 왕자 스토리를 친구들과 공유하고 내용을 추가하거나 수정해 보자.

협력하여 개발하기 친구들과 함께 스토리를 구현할 때 다음 조건을 지키도록 하자.

딱 프로그래밍 조건



- 한 명당 한 번에 5개의 블록만 사용할 수 있다.
- 한 번의 기회에 3분을 초과할 수 없다.
- 다른 친구가 구현한 블록을 수정하거나 삭제할 수 있다.

피드백하기

우리 팀이 구현한 어린 왕자 스토리를 발표하고 친구들의 피드백을 받아 보자.

잘된 점	보완해야 할 점

128 • Ⅲ부 문제 해결과 프로그래밍

참/고/자/료



• 딱 프로그래밍 학습 시 규칙 정하기

- ▶ 서로 신뢰하고 코드에 대해 지나치게 소유욕을 갖지 않기
- ▶ 방어적이지 말고 오픈, 긍정적, 칭찬의 말만 하기
- ▶ 대화를 자주 나누고 무조건 질문하기: 논쟁하여 승자와 패자로 나누지 말 것.
- ▶ 디버깅 같이 하기

• 이야기 만들기 순서

장면 선택하기	캐릭터 만들기	대화 구성하기	디버깅하기
이야기 재구성하기 만들고 싶은 장면 정하기	캐릭터의 움직임을 제어하기 위해 '추가 블록' 만들기 기능 활용	타이밍과 방송하기를 이용해서 스프라이트 사이의 상호 관계를 조직화하기	테스팅을 통해 프로그램의 오류 찾고, 수정하기

• 딱 프로그래밍 환경 구성

- ▶ 컴퓨터 1대에 2명이 앉아서 서로 교대하며 작성하는 방법
- ▶ 컴퓨터 1대에 마우스와 키보드를 2개씩 USB에 꽂아 이동 없이 작성하는 방법
- ▶ 빔 프로젝터나 2대의 모니터를 이용하여 같은 프로그래밍을 고민하여 작성하는 방법 등

182 • Ⅲ부 문제 해결과 프로그래밍

프로젝트 3

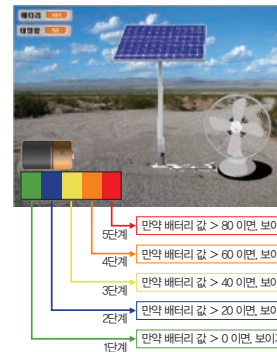
스마트 선풍기



..... 예시 소스 p. 184~p. 185



배터리의 충전 표시



탐구하기

1 문제 상황의 목표 상태를 적어 보자.

현재 상태	시동이 꺼지면 전기를 사용할 수 없어 선풍기를 사용할 수 없다.
목표 상태	

2 태양열을 이용한 스마트 선풍기를 작동시키기 위한 기능들을 분해하여 보았다. 빈칸에 알맞은 단 어들을 보기에서 골라 적어 보자.

보기	태양광이 일정한 <input type="text"/> 가 되면 <input type="text"/> 에 <input type="text"/> 가 충전된다.
전기, 배터리, 세기, 선풍기	<input type="text"/> 에 충전된 <input type="text"/> 로 <input type="text"/> 를 돌린다.
	<input type="text"/> 가 작동되면 충전된 <input type="text"/> 가 사용된다.

Ⅲ-3장 프로그래밍 • 129

27 차시  
소단원 지도 GUIDE

3-5

프로젝트 3

스마트 선풍기

• D-D-D 전략(개발 중심 모형)

개발 중심 모형은 소프트웨어 공학적인 측면에서 SW 개발의 전 과정을 이해하는 모형이다. 개발하고자 하는 소프트웨어에 대한 탐구(Discovery)를 통해 지식을 구성하도록 하고, 개발을 위한 기초를 설계(Design)하도록 하며, 디버깅을 강조하여 소프트웨어를 개발(Development)하도록 지도한다.

탐구 (Discovery)	기능의 이해, 분석, 탐색, 구현
설계 (Design)	문제 이해, 설계, 계획
개발 (Development)	구현, 공유, 개발, 산출

• DDD 모형을 적용하여 탐구와 설계 과정을 거쳐 개발하도록 한다.

탐구하기 답안

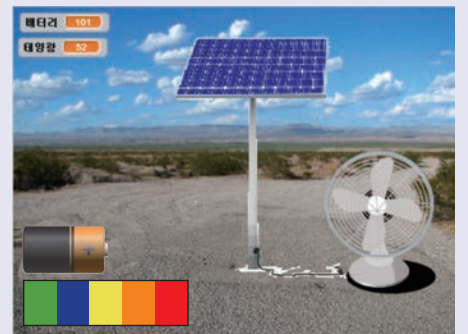
- 1 목표 상태: 태양열에 의해 차 안에서 선풍기를 작동시킬 수 있다.
- 2 세기, 배터리, 전기, 배터리, 전기, 선풍기 선풍기, 전기

• 탐구하기

- ▶ 문제 상황에 대한 현재 상태와 목표 상태를 알아본다.
- ▶ 태양열을 이용한 스마트 선풍기를 작동시키기 위한 기능을 분해 과정을 완성해 본다.

• 유의점

- ▶ 일상생활에서 접하게 되는 문제를 효율적으로 해결하기 위해서 문제 상황을 이해하고, 해결 가능한 형태로 표현한 후 프로그램으로 구현하여 해결해 본다.
- ▶ 만화를 보고 문제 상황을 이해하고 태양광을 충전하여 사용하는 상황들에 대해 이야기해 보며 공감대를 높인다.



Ⅲ-3장 프로그래밍 • 183



설계하기

- ▶ 핵심 요소의 세부 조건들을 추상화해 본다.
- ▶ 스마트 선풍기를 구현하기 위한 알고리즘을 완성해 본다.

설계하기

답안

- 3 배터리 충전: 증가시킨다.  
배터리 상태: 방전, 충전 중  
선풍기의 회전: 감소시킨다.
- 4 >, 배터리  
<, 충전, 방전  
>, 선풍기, 배터리

구현하기

3가지 핵심 요소에 대한 알고리즘을 이용하여 프로그램을 개발해 본다.

유의점

혼자 해결하기 힘든 학생들은 둘 이상의 학생이 서로 협력하여 탐구하고 개발한다.

설계하기

- 3 각 핵심 요소의 작동 조건들을 괄호 안 단어 중 하나를 선택하여 완성시켜 보자.
- 4 스마트 선풍기를 구현하기 위한 알고리즘을 완성해 보자.

핵심 요소	작동 조건	알고리즘
1 배터리 충전	 태양광의 세기: 1~100 사이의 무작위 수 만약 태양광의 세기 > 50 이면, 배터리 값을 (증가시킨다, 감소시킨다).	태양광의 세기: 1~100 사이 난수 값 태양광의 세기 50 ? (예) 배터리 값 ← 배터리 값 + 2
2 배터리의 상태	 만약 배터리 값 < 5 이라면, "방전, 충전 중"이라고 표시한다. 아니면, "방전, 충전 중"이라고 표시한다.	배터리 값 5 ? (예) "중입니다." 메시지 보이기 (아니오) " "입니다." 메시지 보이기
3 선풍기의 회전	 만약 배터리 값 > 30 이라면, 선풍기를 돌린다. 배터리 값을 (감소시킨다, 증가시킨다).	배터리 값 30 ? (예) 선풍기 돌기 속도 ← 선풍기 돌기 속도 + 30 배터리 값 ← 배터리 값 - 1

구현하기

- 5 3가지 핵심 요소에 대한 알고리즘을 이용하여 프로그램을 개발해 보자.

사용 스프라이트

태양광 패널, 선풍기, 배터리

핵심 요소	명령 블록
1 배터리 충전	 만약 라면 태양광 을(를) 0 로 정하기 배터리 을(를) 1 만큼 바꾸기 태양광 을(를) 1 부터 100 사이의 난수
2 배터리의 상태	 만약 라면 배터리 > 5 알하기
3 선풍기의 회전	 만약 라면 선풍기속도 을(를) 1 만큼 바꾸기 배터리 을(를) 1 만큼 바꾸기 선풍기속도 도 돌기 선풍기속도 을(를) 1 만큼 바꾸기

130 • Ⅲ부 문제 해결과 프로그래밍

되돌아보기

- 만든 프로그램을 다른 친구들과 공유하여 개선 방법을 논의해 보자.
- 프로그램이 구동은 되나 자신의 생각대로 작동되지 않는 경우 디버깅 과정을 통해 오류를 수정해 보자.

- 선풍기가 제대로 작동하였는가? 그렇지 않다면 이유는 무엇인가?

- 선풍기 날개의 회전 방향을 반대로 작동시키려면 어떻게 해야 할까?

- 충전 상태를 그림으로 나타내는 스크립트를 작성해 보자. 그리고 태양광 스마트 선풍기의 기능을 조금 더 보완하기 위하여 여러 가지 기능을 추가시켜 수정하여 보자.

- 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

Ⅲ-3장 프로그래밍 • 131

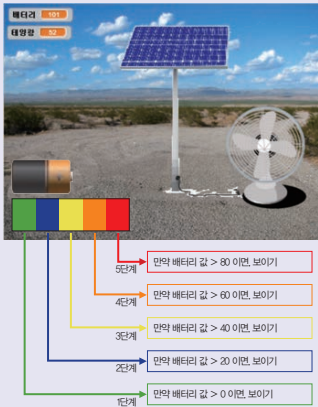
238쪽 디자인 저널(부록 1) 쓰기

- ▶ 되돌아보기의 질문에 답하고 이를 통해 자신의 프로젝트를 수정·보완해 본다.
- ▶ '디자인 저널'은 프로젝트를 수행하는 과정에 대한 학생들의 생각, 새롭게 알게 된 점, 느낀 점 등을 자유롭게 기술하게 함으로써 학생들의 컴퓨팅 사고력의 발전을 평가할 수 있다.

평가

- ▶ 친구들과 공유하여 개선 방안을 찾아도록 한다.
- ▶ 개선해야 하는 부분을 디버깅하도록 한다.

배터리의 충전 표시(교과서 129쪽)

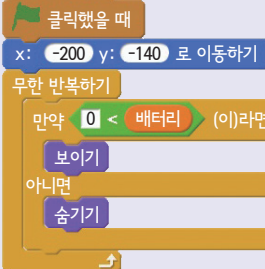


참/고/자/료

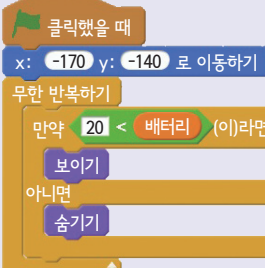
프로젝트 3 스마트 선풍기 예시 소스

배터리의 충전 표시 소스

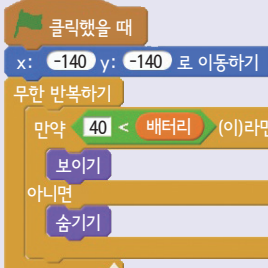
배터리 1단계



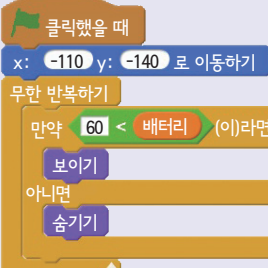
배터리 2단계



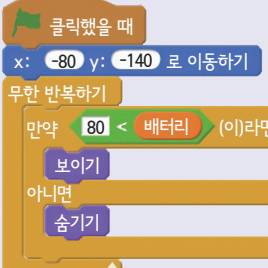
배터리 3단계



배터리 4단계

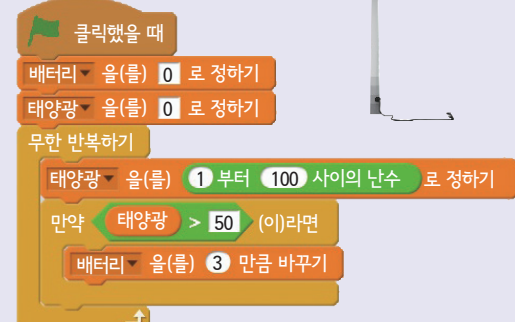


배터리 5단계



핵심 요소

1 배터리 충전  
태양광 패널(메인 소스)



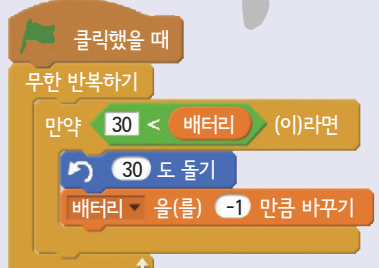
핵심 요소

2 배터리의 상태  
배터리



핵심 요소

3 선풍기의 회전  
프로펠러



프로젝트 4

자동 문자 응답  
시뮬레이터

• N-D-I-S 전략(디자인 중심 모형)

디자인 사고 과정은 탐색을 통한 물입의 과정을 통한 인간 중심의 요구 분석을 의도한다. SW 개발이 단지 기계적인 프로그램 제품 개발이 아닌 인간의 삶을 개선하고 인류의 안전과 요구에 부합하는 활동임을 인식하여 창의적 설계를 진행하도록 한다. 컴퓨팅 사고를 신장시키기 위한 설계와 개발의 과정을 통해 프로토타입 또는 시뮬레이션을 제작하도록 한다. 또한 개발된 결과에 대한 공유와 평가를 통해 사용자 요구에 따른 개선 방법을 찾도록 지도한다.

요구 분석 (Needs)	문제 이해, 인간 중심 요구 분석
디자인 (Design)	창의적 설계, 계획, 공학적 설계
구현 (Implementation)	개발 및 구현, EPL, 피지컬 컴퓨팅

요구 분석하기

답안

- 1 현재 상태: 수업 시간에 온 문자를 확인할 수 없고 그때마다 응답 문자를 보낼 수도 없다.
- 2 목표 상태: 수업 시간이나 공부 시간에 응답해야 할 문자가 온다면 그에 적절하게 응답 문자를 보내 준다.

프로젝트 4

자동 문자 응답 시뮬레이터



수업 시간이나 공부하는 시간에 올리는 문자 때문에 여간 신경 쓰이는 게 아니다. 상대방에게 응답을 해야 할 것 같지만 응답을 하려다가 집중하던 공부에 지장을 줄 수 있기 때문이다. 하지만 상대방이 급하게 연락한 경우라면 내 상황에 대한 문자 응답을 자동으로 보내려 한다.

요구 분석하기

- 1 문제 상황과 목표 상태를 적어 보자.

현재 상태	1
목표 상태	2

- 2 아래와 같이 내게 수신된 문자들을 수집해 보자. 그리고 어떤 문자에 답장을 보내야 하는지 그 번호들의 패턴을 찾아 보자.

친구, 가족들의 휴대전화 번호 010-***-****문자	연락처에 저장되어 있지 않지만 아는 사람 휴대전화 번호 010-***-****	지역 전화번호로 시작되는 학교 등의 연락문자	지역 전화 번호로 시작하는 홍보 문자
명진아 학교 끝났니? 연락 줘~	선영아, ** 초등학교 동문회 있다~ 00월 00일 00시 **초등학교 알 분석점	안녕하세요? 기말 시험이 25일이라서 이번 주말 학원 수강 있습니다.	*** 도서관 00월 00일 저녁 7시 반 <문화가 발견한 예술 인문학> 관심 부탁 드립니다
인터넷 전화 070-***-****로 시작하는 스캠 번호 문자	15**-****로 시작하는 광고성 문자	114, 112 등의 공공 번호 문자	
△△△ 클리닉 명절 이벤트 선착순 00명 20% 할인	[**은행 상품 안내] 주거래 적금 특별금리우대 이벤트 진행중! (00월 00일까지)	119에서 긴급 구조를 위해 귀하의 위치를 확인하였습니다	

요구 분석하기

- ▶ 글에서 제시된 문제 상황의 목표 상태를 적어 본다.
- ▶ 수신 문자를 수집하고 어떤 문자에 답장을 보내야 하는지 수신 번호들의 패턴을 찾아본다.
- ▶ 응답 문자를 보내야 하는 번호와 그렇지 않은 번호를 구분하는 기준을 이야기한다.

• 유의점

실제 학생들이 최근에 받은 문자들을 보며 부재중 응답 문자를 보내야 하는 문자와 그렇지 않은 문자들로 분류해 본다.

• 구현하기

실행 화면과 주요 스크립트를 참고하여 자동 문자 응답 시뮬레이터를 제작해 본다.

• 유의점

핵심 요소별 기능에 대한 추상화와 알고리즘을 연결하여 논리적으로 일관성 있게 완성해 본다.

- 3 내 연락처에 저장되어 있는 번호인지 아닌지, 010으로 시작하는 번호인지 아닌지

- 3 응답 문자를 보내야 하는 번호와 그렇지 않은 번호를 구분하는 기준은 무엇일까?

설계하기

- 4 자동 문자 응답 프로그램을 구현하기 위해 다음 사항을 생각해 보자.

핵심 요소	<input type="text"/> 문자 번호, <input type="text"/> 에 저장된 번호	
패턴 찾기	응답 문자를 보내야 하는 번호의 패턴: 번호 앞 3자리가 <input type="text"/> 으로 시작함	
추상화	리스트가 필요한 항목	<input type="text"/> , <input type="text"/>
	변수가 필요한 항목	<input type="text"/>
	① 수신 문자의 수신 번호 = <input type="text"/> : 누구인지 되묻는 문자 ② 수신 문자의 수신 번호 = <input type="text"/> : 지인에게 부재중 문자	

- 5 자동 문자 전송 앱의 작동 순서를 세부적으로 나누어 알고리즘을 적어 보자.

- ① 문자 수신 → 수신 번호의 ( ) = 1 and ( ) = 0 ?
- ② (yes) → ( )
- ③ (no) → 스캠 문자로 인식
- ④ 수신 번호 = ( ) ?
- ⑤ (yes) → ( )
- ⑥ (no) → 무응답 → ①행 반복 시작

구현하기

- 6 실행 화면과 주요 스크립트를 참고하여 자동 문자 응답 프로그램의 시뮬레이터를 제작해 보자.



설계하기

- ▶ 자동 문자 응답 시뮬레이터의 핵심 요소, 패턴, 추상화 과정을 완성해 본다.
- ▶ 자동 문자 전송 앱의 작동 순서를 세부적으로 나누어 알고리즘을 채워 보도록 한다.

설계하기

답안

- 4 수신, 연락처 010  
연락처 목록, 응답할 문자 목록  
수신 문자 번호 010  
연락처 목록
- 5 둘째 자리, 셋째 자리  
누구인지 되묻는 문자 전송  
연락처 목록  
부재중 문자 전송

• 구현 코드 예시 •

자동 응답 문자 시뮬레이터 만들기



• 혼자 해결하기 힘든 학생들은 둘 이상의 학생이 서로 협력하여 탐구하고 개발하도록 한다.

• 공유하기  
다른 친구들과 공유하여 개선 방법을 논의해 본다.

• 디자인 저널 쓰기  
되돌아보기의 질문에 답하고 이를 통해 자신의 프로젝트를 수정·보완한다.

‘디자인 저널’은 프로젝트를 수행하는 과정에 대한 학생들의 생각, 새롭게 알게 된 점, 느낀 점 등을 자유롭게 기술하게 함으로써 학생들의 컴퓨팅 사고력의 발전을 평가할 수 있다.

공유하기 자신이 만든 프로그램을 다른 친구들과 공유하여 개선 방법을 논의해 보자.

잘된 점	보완해야 할 점

#### 되돌아보기

• 프로그램이 구동은 되나 자신의 생각대로 작동되지 않는 경우 디버깅 과정을 통해 오류를 수정해 보자.

• 리스트와 변수의 차이는 무엇일까? 전화번호의 값을 리스트 대신 변수를 사용하여 구현이 가능할까?

• 실행해서 오류가 생기는 경우는 언제인가? 어떻게 해결해야 하는가?

• 이번 활동을 통해 생각하거나 느낀 점을 자유롭게 써 보자.

### 컴퓨터 과학자처럼 생각하기

컴퓨터 과학자들은 프로그램을 개발할 때 아래의 사고 과정을 중요하게 생각한다.  
여러분도 앞의 4가지 프로젝트를 개발하면서 컴퓨터 과학자처럼 생각했는지 확인해 보자.

#### 실험하기와 반복하기

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
프로젝트를 작은 부분에서 시작하여 점차 확대하여 만들었는가?			
프로젝트의 일부만이 제대로 실행되는지 확인하였는가?			
프로젝트 실행 결과에 따라 다양하게 수정하여 보았는가?			
프로젝트를 다른 방법으로 시도하거나 새로운 방법으로 시도하였는가?			

#### 시험하기와 수정하기

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
프로젝트를 실행하였을 때 결과를 다양하게 시험하였는가?			
원하는 실행 결과와 다른 점을 발견하였는가?			
문제의 원인을 찾기 위해 스크립트를 분석하였는가?			
오류가 발생한 결과를 확인하고 수정하였는가?			

#### 재사용과 재구성

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
다른 사람들이 만든 프로젝트를 실행하여 아이디어를 찾았는가?			
다른 프로젝트에서 좋은 부분을 선택하여 내 프로젝트에 적용하였는가?			
프로젝트를 개선하기 위해 기존의 프로젝트를 수정하였는가?			
프로젝트에 참고한 작품이나 아이디어를 얻은 작품의 출처를 밝혔는가?			

#### 추상화와 모듈화

컴퓨터 과학자처럼 생각하기	많이 함	보통	적게 함
프로젝트에 필요한 스프라이트를 어디에 사용할지 결정하였는가?			
프로젝트에 필요한 핵심 스크립트를 구현하였는가?			
다른 사람들이 이해하기 쉽게 스크립트를 개발하였는가?			

### 컴퓨터 과학자처럼 생각하기

- ▶ ‘실험과 반복하기’, ‘시험하기와 수정하기’, ‘재사용과 재구성’, ‘추상화와 모듈화’의 4가지 영역에 대하여 학생들이 구현한 프로젝트에 대해서 구체적으로 설명하도록 함으로써 학생들의 컴퓨팅 사고력의 실습 영역에 대한 기술형 수행 평가를 할 수 있다.
- ▶ ‘CT 실습(practice)에 대한 수행 평가 기준’에 따라 평가한다.

#### 구현 코드 예시

리스트에 테스트용 연락처 목록과 응답 문자 목록을 넣기

클릭했을 때

모두▼ 번째 항목을 연락처 목록▼ 에서 삭제하기

01022293587 항목을 연락처 목록▼ 에 추가하기

01035461234 항목을 연락처 목록▼ 에 추가하기

01024597896 항목을 연락처 목록▼ 에 추가하기

0102417577 항목을 연락처 목록▼ 에 추가하기

클릭했을 때

모두▼ 번째 항목을 응답문자 목록▼ 에서 삭제하기

“공부중이에요. 쉬는 시간에 연락드릴게요.” 항목을 응답문자 목록▼ 에 추가하기

“수업중입니다. 누구인지 남겨주시면 연락드리겠습니다.” 항목을 응답문자 목록▼ 에 추가하기

#### CT 실습(practice)에 대한 수행 평가 기준(brennan, 2012)

영역	낮음	보통	높음
실험과 반복하기	새롭게 시도한 예시를 제시하지 못했다.	새롭게 시도한 예시를 제시하였으나 적용 방법은 기술하지 못했다.	새롭게 시도한 예시와 적용 방법을 구체적으로 기술하였다.
시험하기와 수정하기	문제 상황을 제시하지 못했다.	디버깅한 문제 상황을 제시하였으나 해결 방법은 기술하지 못했다.	디버깅한 문제 상황과 해결 방법을 구체적으로 기술하였다.
재사용과 재구성	재구성한 다른 프로젝트를 제시하지 못했다.	재구성한 다른 프로젝트의 일부분을 제시하였으나 구체적인 구현 방법은 기술하지 못했다.	재구성한 다른 프로젝트의 일부분과 구현 방법을 구체적으로 기술하였다.
추상화와 모듈화	추상화한 동작이나 상황 모듈을 제시하지 못했다.	추상화한 동작이나 상황 모듈을 제시하였으나 구체적인 스크립트는 기술하지 못했다.	추상화한 동작이나 상황 모듈을 제시하고 구체적인 스크립트도 기술하였다.



**문제 상황:** 주영은 미술 수행 평가로 꽃을 주제로 한 인포그래픽을 만들기로 했다. 주변에 피어 있는 꽃을 관찰해 보니 대부분의 꽃잎 수가 3장, 5장, 8장, 13장으로 되어 있었다. 무엇보다 놀라운 점은 이런 규칙성이 꽃잎의 패턴뿐만 아니라 솔방울 포엽, 파인애플의 껍질 등 자연의 도처에서 나타난다는 점이다. 또한 달팽이 껍데기뿐만 아니라 독수리가 나선을 그리면서 비행하는 모습이 꽃잎의 패턴과 유사하다는 것이다. 자연의 현상을 수학적으로 설명할 수 있다니, 오~ 흥미로운 사실이다. 이러한 규칙을 이용해서 독수리가 날아가는 모양을 그려 보기로 했다.

### 1 문제 이해하고 분석하기

#### ① 문제 이해

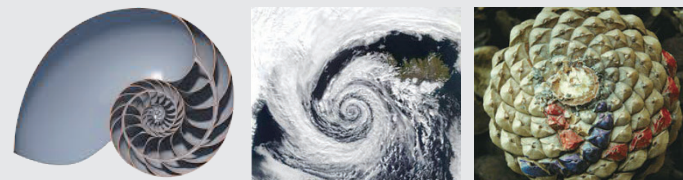
식물 이외에 앵무조개 껍질의 무늬에도 유사한 규칙이 있고, 달팽이 껍질과 여러 바다 생물의 껍질에서도 이러한 규칙을 발견할 수 있다.

다음 그림에서 발견된 규칙이 있는 그림을 구현해 보자.



#### ② 문제 분석

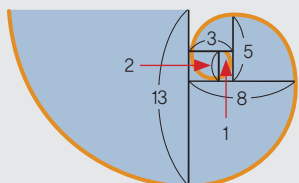
앵무조개와 태풍, 솔방울의 무늬에서 나타나는 공통적인 모양을 관찰해 보자. 각 무늬에서 나타나는 공통적인 모양은 어떠한가? 공통적인 모양이 가진 규칙을 찾아보고, 피보나치의 수와 일치하는 정도를 확인해 보자.



공통적으로 나타나는 나선형 구조를 확인한다. 나선형의 중심으로부터 일정한 간격으로 나누어 반지름의 길이를 측정한다. 90도 간격으로 나눈 반지름의 길이를 측정한다.

### 2 문제 해결 방법의 설계

#### ① 핵심 요소 찾기 및 추상화



오브젝트	연필
핵심 요소	(중심점): 나선형이 시작되는 지점 (회전하는 각도): 나선형의 궤적을 나타내는 각도 (반지름 ): 중심점과의 거리
패턴 찾기	90도 간격으로 반지름의 길이가 일정하게 증가한다. 증가하는 값은 90도 전의 반지름 길이를 합한 수이다.
추상화	피보나치 수는 바로 앞의 두 피보나치 수의 합이 된다.

#### ② 문제 분석

문제를 해결하기 위해 수행되는 일련의 단계를 설계해 보자.

- 나선형의 중심이 될 좌표를 선정한다.
- ‘center’, ‘dot’ 로 두 개의 오브젝트는 작은 점이다.
- ‘center’ 오브젝트는 부채꼴의 중심에 위치하고 현재 위쪽 방향을 보고 있다.
- ‘dot’ 오브젝트는 중심에서 1만큼 떨어진 위치로 이동하여 점을 찍는다.
- 이런 동작을 90번 하게 되면 점으로 이루어진 호가 생기게 된다.

### 3 문제 해결 방법의 구현

1. 스프라이트 ‘center’ 의 스크립트	2. 스프라이트 ‘dot’ 의 스크립트
<pre> 클릭했을 때 지우기 temp 을(를) 0 로 정하기 a 을(를) 1 로 정하기 b 을(를) 1 로 정하기 x: 0 y: 0 로 이동하기 0 도 방향 보기 무한 반복하기   90 번 반복하기     draw 발송하기     1 도 돌기   -1 + a 만큼 움직이기   temp 을(를) a 로 정하기   a 을(를) b 로 정하기   b 을(를) temp + a 로 정하기         </pre>	<pre> 클릭했을 때 펜 색깔을 (으)로 정하기 펜 굵기를 (으)로 정하기 draw 을(를) 받았을 때   펜 올리기   펜 명암을 10 만큼 바꾸기   center 위치로 이동하기   방향 of center 도 방향 보기   b 만큼 움직이기         </pre>
‘center’ 스프라이트는 처음 무대의 중심에 위치하고 위쪽 방향을 향한 상태에서 시계 방향으로 1도씩 회전한다.	매번 ‘draw’라는 메시지를 받을 때마다 점을 찍어 준다.

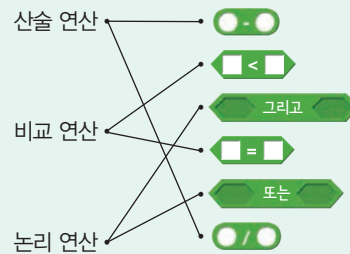
# 배운 내용 정리 & 평가하기

## 평가하기

1 프로그래머, 프로그래밍, 프로그래밍 언어

2 순차, 조건, 반복

3



## 참고자료

### 세계 경제 포럼과 4차 산업 혁명

세계 경제 포럼은 1971년 독일의 경제학자 클라우스 슈밥이 세운 유럽경영포럼(European Management Forum)으로 출발했다. 1981년부터 스위스의 휴양 도시 다보스에서 매년 회의가 열리기 때문에 '다보스 포럼'이라고 불리기도 한다. 세계 각국의 정상, 장관, 국제 기구 수장, 재계 및 금융계 최고 경영자들이 모여 각종 정보를 교환하고, 세계 경제 발전 방안 등에 대해 논의하는 자리로 세계 경제에 미치는 영향이 큰 포럼이다. 이 포럼에서 2016년 최고의 논의 과제는 단연 '4차 산업 혁명'이었다.

클라우스 슈밥은 이 자리에서 '4차 산업 혁명'을 이끌 메가트렌드(megatrend)로서 '물리학(physical) 기술', '디지털(digital) 기술',

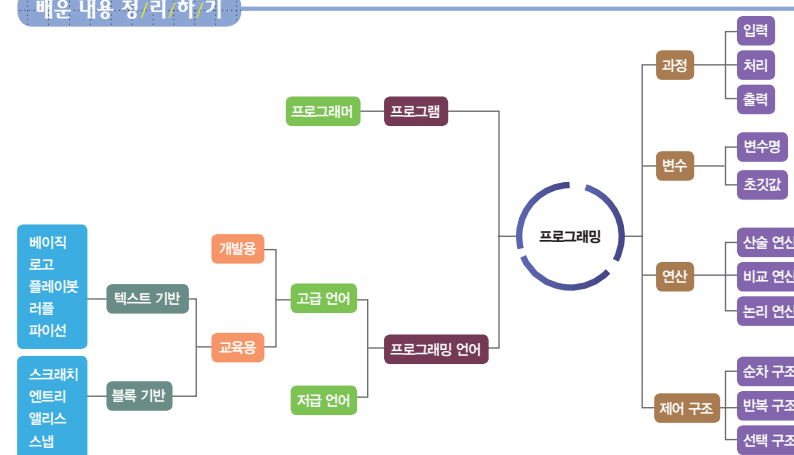
'생물학(biological) 기술'을 꼽았다. 2025년을 4차 산업 혁명의 티핑포인트로 설정하고 다음과 같은 다양한 기술이 실현될 가능성에 대해 제시하였다.

체내 삽입형 기기, 디지털 정체성, 웨어러블(wearable) 인터넷, 유비쿼터스 컴퓨팅, 주머니 속 슈퍼컴퓨터, 사물 인터넷, 커넥티드 홈, 스마트 도시, 빅 데이터를 활용한 의사 결정, 자율 주행 자동차, 인공지능을 통한 의사 결정, 로봇을 이용한 서비스, 비트코인과 같은 블록체인 화폐의 확대, 공유 경제, 3D 프린팅 기술에 의한 제조업 혁명, 맞춤형 아기 등이 그 예이다.



▲ 클라우스 슈밥

## 배운 내용 정리하기



## 평가하기

1 빈칸에 들어갈 용어를 쓰시오.

소프트웨어는 컴퓨터 프로그래밍이라고도 한다. 따라서 소프트웨어는 인간의 언어가 아닌 컴퓨터가 알아들을 수 있는 컴퓨터 언어로 만들어져야 한다. 컴퓨터 언어로 소프트웨어를 만드는 사람을 ( ), 소프트웨어를 만드는 작업을 ( )이라 한다. 그래서 이 프로그래밍에 사용되는 컴퓨터 언어는 ( )라고 한다.

2 빈칸에 들어갈 용어를 쓰시오.

프로그램의 명령 실행 구조는 순차, 반복, 선택 구조이다. ( ) 구조는 하나의 방향으로 진행되는 것을 말하고, ( ) 구조는 주어진 조건에 따라 실행 결과가 달라진다. ( ) 구조는 주어진 조건에 따라 특정 명령어를 반복해서 실행한다.

3 스크래치 프로그램에서 사용되는 연산 블록을 유형에 맞게 연결하시오.

- 산술 연산 • +, -, \*, /, %
- 비교 연산 • <, >, <=, >=, ==, !=
- 논리 연산 • and, or, not

136 • 3부 문제 해결과 프로그래밍

## 3부 마무리

## 단원 평가

## 단원 되돌아보기

## 3부

## 단원 평가 정답

1 다음 문제 상황을 읽고 다음 문제를 해결해 보자.

### 스마트 스프링클러를 이용한 땅의 수분 측정하기

A 학생 가족은 주말마다 농장에 가서 감자와 고구마 등을 심어서 가꾸고 있다. 주말밖에 갈 시간이 없어서 주중에 물을 주기 어렵다. 주중에도 땅이 마르거나 날이 더우면 자동으로 물을 주는 스프링클러가 필요하다.

① 위 문제 상황의 목표 상태는 무엇인가?

땅이 마르거나 날이 더우면 자동으로 물을 주는 스프링클러 제어 프로그램

② 위 문제를 해결하기 위하여 다음과 같이 땅의 수분 정도에 따른 센서 저항을 측정해보았다.

- 1 6개의 종이컵에 마른 흙을 담고 물을 조금씩 다르게 넣는다.
- 2 저항 양쪽을 흙에 꽂아 흙의 수분에 대한 저항 값을 측정한다.

손으로 만졌을 경우 흙의 수분 정도	0	1	2	3	4	5
센서의 저항 값	98	87	65	35	28	12

스프링클러가 흙의 수분 정도가 0과 1일 때 작동하기 위한 조건식을 부등호를 사용하여 완성해 보시오.

수분 센서의 저항 값 ( > ) 800이면 스프링클러 돌기

[해설] 수분 정도가 0은 센서의 저항 값이 98이고 1은 87이므로 수분 센서의 저항 값이 80보다 크도록 설명하도록 한다.

2 다음과 같이 이진 탐색을 이용하여 1~1,000의 숫자 중에 875를 찾는 과정에서 탐색 기준 값에 해당하는 숫자를 적어 보시오.

탐색 횟수	0회	1회	2회	3회
탐색 기준	1000	500	750	875

[해설] 이진 탐색은 중간 값을 기준으로 반으로 나누어 탐색하는 방식이므로 각 구역의 중간 값이 기준 값이 된다.

3 아래 동전을 이용하여 2,980원을 지불하기 위해서 필요한 가장 적은 동전의 개수는 몇 개인가?

예를 들어 아래의 동전들로 1,050원을 지불할 때를 생각해 보자. 1,050원을 지불하는 가장 적은 동전 수는 500원짜리 2개와 50원짜리 1개를 사용하여 총 3개의 동전을 이용하는 것이다.



빈칸에 알맞은 숫자를 적어보시오.

	500원	100원	50원	10원	계
동전 필요 개수	5	4	1	3	13

[해설] 2,980원을 지불하는 최소의 동전 개수는 큰 동전부터 빼는 방식으로 하여 500원 5개(2,000원), 100원 4개(400원), 50원 1개(50원), 10원 3개(30원)이므로 총 13개이다.

3부 마무리 단원 평가

137

## 단원 평가

## 단원 되돌아보기

4 은아, 찬민, 지민이가 학교를 마치고 어디로 놀러갈지를 이야기하고 있다. 주사위 3개를 순서대로 던져 아래와 같은 규칙으로 놀러 가려고 한다.

- 1 만약 첫 번째 주사의 눈이 두 번째 주사위 눈보다 작은가?  
2     **그러면:** 강으로 놀러 간다.  
3     **아니면:** 만약 세 번째 주사의 눈이 첫 번째 보다 작은가?  
4         **그러면:** 산으로 놀러 간다.  
5         **아니면:** 놀이터로 놀러 간다.



놀이터로 놀러가게 되는 주사위의 조건을 부등호로 나타내 보고 예시를 하나 적으시오.

첫 번째 주사위 눈 (  $\geq$  ) 두 번째 주사위 눈  
세 번째 주사위 눈 (  $\geq$  ) 첫 번째 주사위 눈  
주사위 눈의 예 : ( 3 ) - ( 2 ) - ( 4 )

[해설] 놀이터로 놀러가기 위해서는 두 가지 조건 모두 '아니요'가 나와야 한다. 첫 번째 주사위 눈은 두 번째 주사위 눈보다 작지 않아야 하므로 크거나 같아야 하고, 세 번째 주사위 눈은 첫 번째 주사위보다 작지 않아야 하므로 크거나 같아야 한다.

5 다음 연산을 수식으로 표현하고, 그 결과 값을 계산해 보자. 연산의 결과 값이 다르게 나타난다면 왜 그런지 이유를 살펴 보자.

연산 스크립트	수식	결과값
	$20 + (5 \times 3) - (8 \div 2)$ $20 + 5 \times 3 - 8 \div 2$	31
	$\{20 + (5 \times 3)\} - (8 \div 2)$ $(20 + 5 \times 3) - 8 \div 2$ $(20 + 5 \times 3) - (8 \div 2)$ $\{20 + (5 \times 3)\} - 8 \div 2$	31
	$\{(20 + 5) \times 3\} - (8 \div 2)$ $(20 + 5) \times 3 - 8 \div 2$	71

결과 값이 다르게 나타나는 이유 :

연산자의 우선 순위는 괄호 안의 계산을 가장 먼저해야 하며 곱하기, 나누기 다음에 덧셈과 뺄셈을 해야 하기 때문에 결과 값이 다르게 나타날 수 있다.

[해설] 위의 수식으로 표현한 것과 같이 연산자 블록의 삽입 위치에 따라 연산 순서가 달라지므로 그 결과 값도 달라질 수 있다.

6 다음 보기처럼 연필이 중심에서 네 개의 같은 정사각형을 그릴 수 있도록 스크립트의 빈칸을 완성하시오.

[정답] 4, 4, 90, 90

[해설] 100만큼 움직이기와 90도 돌기를 4번 반복하여 작은 사각형을 그리고 방향을 90도 돌려 작은 사각형 그리기를 4번 반복한다.

7 미로에서 고양이가 녹색 사각형에 부딪치지 않고 연두색 사각형까지 가야 한다. 고양이가 녹색 벽을 통과하지 못하도록 프로그램 명령을 추가해 보자.

### 추가해야 할 명령

[예시 답안]

- ① 녹색에 닿으면 뒤로 10만큼 움직인다.  
② 녹색에 닿으면 처음 위치로 이동한다.  
등 스크립트로 그려 넣어도 상관없음