



컴퓨팅사고와 코딩기초

Chapter 5. 알고리즘

목차

- 컴퓨팅사고와 알고리즘
- 알고리즘의 표현
- 의사코드와 순서도 작성



컴퓨팅사고와 알고리즘

❖ 알고리즘이란?

- 어떠한 일을 수행하거나 문제를 해결하기 위한 논리적인 절차의 나열로 일련의 명령이나 규칙의 집합
- 알고리즘은 컴퓨터 프로그램 작성의 기초
- 알고리즘의 설계 없이 곧바로 프로그래밍을 만들면 생각지 못하는 오류사항이나 예외사항이 발생하거나 효과적인 문제 해결에 실패할 수 있음

컴퓨팅사고와 알고리즘

❖ 알고리즘의 조건

입력(input)	알고리즘은 0 이상의 외부에서 제공된 자료가 존재해야 함
출력(output)	알고리즘은 최소 1개 이상의 결과를 가져야 함
명확성(definiteness)	알고리즘의 각 수행 과정은 명확하여 모호하지 않은 명령어로 구성되어야 함
유한성(finiteness)	알고리즘은 단계들을 유한한 횟수로 수행한 후 문제를 해결하고 종료되어야 함
유효성(effectiveness)	알고리즘의 모든 명령들은 명백히 실행 가능한 것이어야 함 (필요시 수작업으로도 검증 가능한 것이어야 함)

컴퓨팅사고와 알고리즘

❖ 알고리즘의 일반적 평가기준

문제에 주어진 기준이나 조건을 충족하는가?

문제와 관련한 모든 입력에 대해 항상 올바른 답을 보장하는가?

동일한 문제 유형에 일반적으로 적용하여 해결할 수 있는가?

기술된 알고리즘이 이해하기 쉬운가?

가능한 한 빨리 최소한의 자원을 사용하여 문제를 해결하는가?

컴퓨팅사고와 알고리즘

❖ 알고리즘의 평가

- 문제 해결의 최선의 방법을 지향해야 한다는 효율성에 초점
- 알고리즘 효율성 평가법
 - 시간 복잡도
 - 공간 복잡도

알고리즘의 표현

❖ 알고리즘의 표현 방법

- 자연어(Natural Language)를 이용한 서술적 표현 방법
- 의사코드(Pseudo Code)를 이용한 추상화 방법
- 순서도(Flow Chart)를 이용한 도식화 표현 방법

알고리즘의 표현

❖ 의사코드(Pseudo Code)를 이용한 표현

- 프로그래밍 언어의 일반적인 형태와 유사한 의사코드 또는 가상 코드 사용
- 특정 프로그래밍 언어가 아니므로 직접 실행은 불가능
- 실제 프로그래밍 언어처럼 엄밀한 문법을 따를 필요가 없기 때문에 다양한 변종 존재

알고리즘의 표현

❖ 의사코드(Pseudo Code)를 이용한 표현

- 의사코드에서 많이 쓰이는 요소들의 예시

기호(또는 문자)		변수명, 함수명, 레코드 필드명, 문장의 레이블 등 나타냄
대입문(Assignment)		\leftarrow , $=$, $:=$
연산자	산술 연산	$+$, $-$, $*$, $/$, mod
	관계 연산	$==(=)$, \neq , $<$, $>$, \leq , \geq
	논리 연산	or, and
제어문	조건 제어	if, than, else, else if
	반복 제어	for, while, repeat-until
그 외 자주 쓰는 키워드		get(입력), print(출력), set(초기화), begin(시작), end(끝) 등




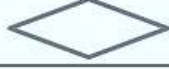





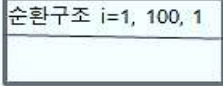
알고리즘의 표현

❖ 순서도(Flow Chart)를 이용한 표현

- 정해진 기호를 사용하여 알고리즘을 표현한 것
- 알고리즘 표현 방법 중 가장 많이 사용되며, 특정 기호를 사용하여 표현하므로 절차나 흐름을 이해하기 쉽고 표현이 명료

알고리즘의 표현

❖ 순서도(Flow Chart)를 이용한 표현

기호	이름	의미
	단자(Terminal)	순서도의 시작과 끝을 표시
	준비(Preparation)	변수의 선언 및 초기값 부여, 배열 선언
	처리(Process)	값을 계산하거나 대입하는 기호
	판단(Decision)	참과 거짓을 판단하거나 조건에 맞는 경로로 분기
	수동입력(Console)	키보드를 이용한 수동입력
	입출력(input/output)	데이터의 입력과 출력
	문서(Document)	처리된 결과를 프린터로 출력
	흐름선(Flow Line)	각종 처리 기호의 처리 흐름을 연결
	연결자(Connector)	다른곳으로의 연결을 표시
	순환구조(Loop)	i가 1부터 100까지 1씩 증가하면서 반복문 내의 명령문 수행. 증가치는 1일때 생략가능

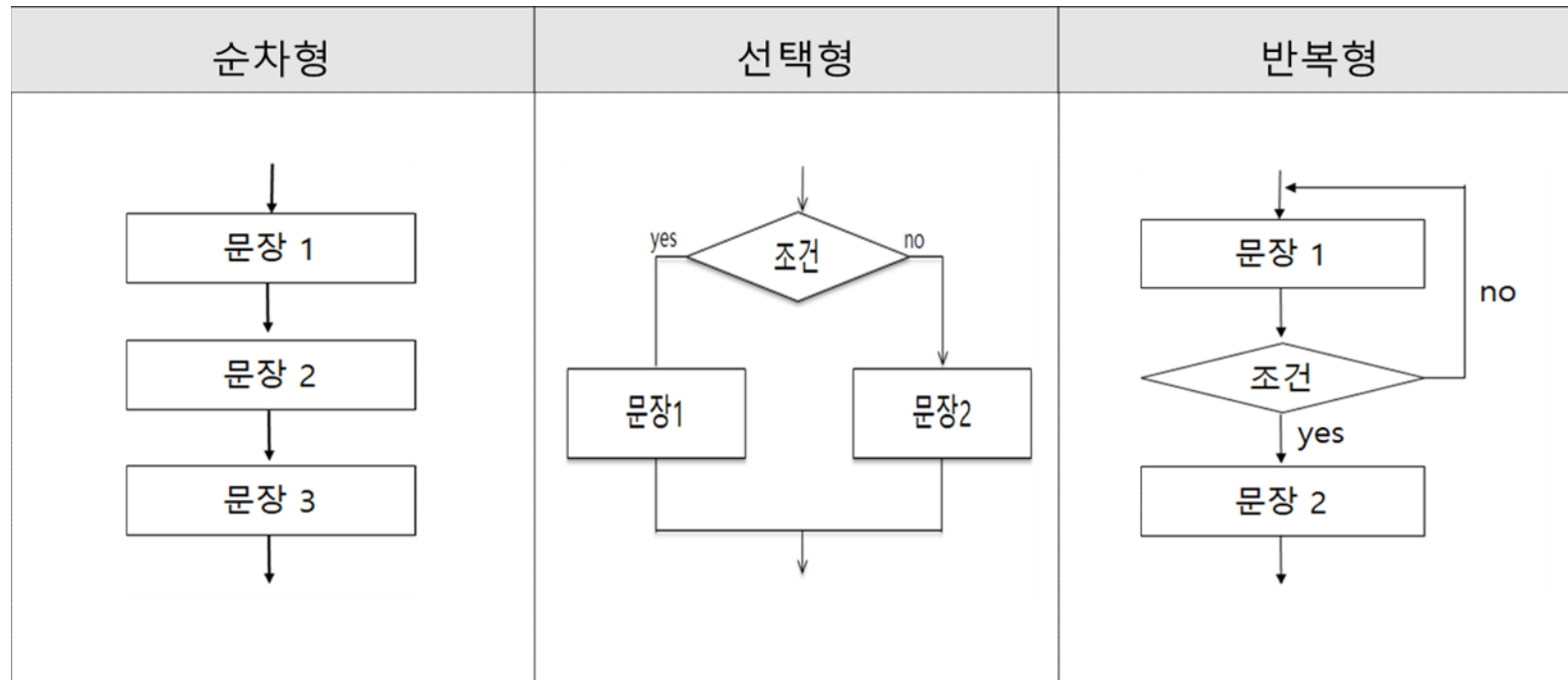
순서도(Flow Chart) 작성

❖ 순서도 작성의 일반적 규칙

- 전체의 흐름을 명확하게 하여 알아볼 수 있도록 작성한다.
- 표준 기호를 사용하여 간단하고 명료하게 표현한다.
- 기호 내부에 처리할 내용을 간략하게 기입한다.
- 위에서 아래로, 왼쪽에서 오른쪽 방향으로 작성한다.
- 과정이 길고 복잡하면 나누어서 작성하고 연결자로 연결한다.

순서도(Flow Chart) 작성

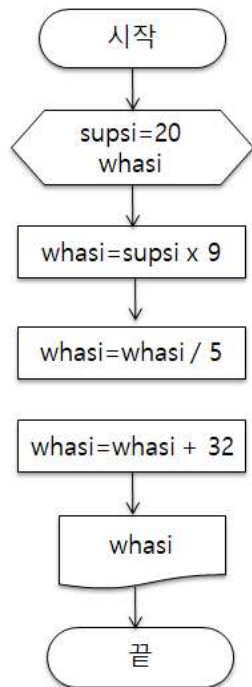
❖ 순서도 작성의 3가지 기본구조



의사코드/순서도 작성

❖ 섭씨온도를 화씨온도로 변환하기

- $whasi = (supsi \times 9) / 5 + 32$

순서도 표현	의사코드 표현
섭씨온도가 20도일 때의 순서도  <pre>graph TD; Start([시작]) --> Input{{supsi=20 whasi}}; Input --> P1[whasi=supsi x 9]; P1 --> P2[whasi=whasi / 5]; P2 --> P3[whasi=whasi + 32]; P3 --> Out[/whasi/]; Out --> End([끝]);</pre>	섭씨온도가 20도일 때의 의사코드 <pre>supsi = 20 whasi = supsi × 9 whasi = whasi / 5 whasi = whasi + 32 print whasi</pre>

