Computational Thinking

2017-2학기

4장-문제 해결

박종혁 교수 (컴퓨터공학과)

jhpark1@seoultech.ac.kr

http://www.parkjonghyuk.net

서울과학기술대학교 SEOULTECH SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

학습개요

- 4장 문제 해결
  1. 문제정의
  2. 논리적 추론
  3. 분해: 소프트웨어 설계
  4. 분해: 다른 사용법들
  5. 추상화: 클래스 다이어그램
  6. 추상화: 유스 케이스 다이어그램

- 조별과제
  - 다음 주 강의 시간 발표

학습목표

- 문제를 잘 해결하기 위한 출발점으로서 문제 정의를 살펴보도록 한다.
- 알고리즘적 문제정의의 핵심인 기능 요구사항들을 탐구한다.
- 소프트웨어 개발에 적용되는, 원인-결과 분석, 연역적 추론, 귀납적 추론과 같은, 논리 추론 방법론들을 알아본다.
- 프로그래밍이라는 것은 알고있는 패턴에 자주 의존하는 활동이라는 것과 제어 흐름의 다섯 가지 패턴은 순차화, 선택, 반복, 제어 추상화, 병렬처리 라는 것을 이해한다.
- 컴퓨팅적인 문제에서 알고리즘의 역할을 이해하고 여러 형태의 알고리즘을 탐색한다.

학습목표

- 개요작성하기와 하향식설계에 유용하게 쓰이는, 문제 해결의 핵심 전략인, 분할정복 방법을 알아본다.
- 이진 탐색의 대안이 되는 선형 탐색과 같이 데이터 분해의 대안이 되는 방법론들의 영향을 생각한다.
- 제어 추상화와 데이터 추상화를 위한 클래스 다이어그램, 문제정의 추상화를 위한 유스 케이스 다이어그램과 같은, 컴퓨터 공학에 중요하게 사용되는 여러 형태의 추상화 기법들를 탐색한다.

컴퓨팅 문제란?

❖ **컴퓨터의 연산능력을 활용하여 해결할 수 있는 문제**

❖ **컴퓨팅 문제의 예**

- 최대 최소 값 찾기
- 주어진 범위에서 소수의 갯수 찾기
- 최단거리 계산
- 기상 예보

컴퓨팅 문제해결을 위한 네 가지 전략

- ❖ 문제정의
- ❖ 논리추론
- ❖ 문제분해
- ❖ 추상화

❖ 소프트웨어가 할 작업들을 기술

- 소프트웨어 개발목표로 작용
- 소프트웨어 개발 후 문제 해결 여부, 동작 여부를 확인할 때 사용
- 문제정의 과정이 생략되면 소프트웨어가 완성된 후
  제대로 동작하는 지를 확인할 수가 없음

❖ 문제정의의 결과물

- 요구사항(requirements)
  - 기능적 요구사항: 소프트웨어가 수행할 작업 관련 부분
  - 비기능적 요구사항: 소프트웨어 관련 특성이나 제약조건 .
    소프트웨어 안전성, 신뢰성, 보안, 성능, 고객지원과 같은 부분
- 잘 작성된 요구사항
  - 분명하고, 일관성 있고, 완전해야 함

❖ **분명해야 한다.**

- F implied N과 같은 논리적 명제로 전환 가능해야 한다.
- "비디오가 재생을 마친다." implied "이 버튼은 동작을 하지 않는다."

❖ **일관성있어야 한다.**

- 요구사항들 간에 모순이 없어야 한다.

❖ **완전해야 한다.**

- 모든 사용 시나리오가 고려되어야 한다.

❖ 요구사항의 예 - 비디오 플레이어

[그림 4.1] 비디오플레이어에 대한 기능적 요구사항 일부

일련번호	V1
이름	재생
행동	플레이 버튼(오른 쪽 그림)을 클릭하면 비디오 재생을 시작한다. 비디오가 임시정지된 경우에는, 비디오 재생을 하게 되면 임시정지된 지점부터 재생을 시작된다. 만약 비디오가 처음부터 재생을 시작한다면, 새로운 컴퓨터 윈도우 창이 떠서 그 창에서 비디오 재생이 시작된다. 비디오가 재생이 되고 있는 상태에는 임시정지/재생 버튼은 임시정지 버튼으로 동작하며 임시정지 버튼 이미지를 화면에 보여준다.

일련번호	V2
이름	임시정지
행동	임시정지 버튼(오른 쪽 그림)을 클릭하면 지금 재생중인 지점에서 재생을 임시정지한다. 비디오가 재생을 마친 상태라면, 이 버튼은 아무 동작도 하지 않는다. 임시정지 버튼을 클릭하게 되면 임시정지/재생 버튼은 재생 버튼으로 동작하게 되며 재생 버튼 이미지를 화면에 보여준다.

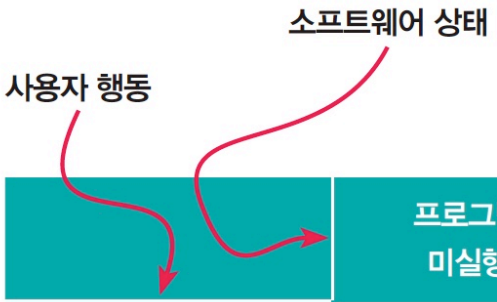
4.1 문제정의

일련번호	V3
이름	음량 증가
행동	현재 소리의 크기가 최대 80dB 레벨이하라면 음량 증가 버튼을 클릭하게 되면 소리의 크기가 5dB 씩 증가한다. 만약 현재 소리의 크기가 80dB라면 아무 행동도 하지 않는다.

일련번호	V4
이름	음량감소
행동	현재 소리의 크기가 조용한 상태 (0dB 레벨)이상이라면 음량 감소 버튼을 클릭하게 되면 소리의 크기가 5dB 씩 감소한다. 만약 현재 소리의 크기가 조용한 상태라면 아무 행동도 하지 않는다.

❖ 요구사항의 완전성 판단

- 상태-활동 테이블 (state-activity table)로 판단함



	프로그램 미실행	비디오 임시정지	비디오 재생	비디오 재생 끝남
재생 클릭		V1		
임시정지 클릭			V2	V2
음량증가 클릭		V3	V3	V3
음량감소 클릭		V4	V4	V4
비디오 파일 더블클릭	V5	V5	V5	V5
윈도우창 더블클릭		V6	V6	V6

그림 4.3 비디오 재생 응용 프로그램의 상태–활동 테이블

❖ 상태 활동 테이블

- 첫 열에는 사용자들의 가능 액션들을 열거한다.
- 첫 행에는 프로그램의 가능한 상태들을 열거한다.
- 각 셀에는 해당하는 요구사항 번호를 적는다.
- 생길 수가 없는 상태에 대해서는 해당 셀을 회색으로 마킹한다.
- 회색 셀도 아니면서 요구사항 번호도 없는 셀은 요구사항이 추가로 필요하다는 것을 의미하므로 이에 맞는 요구사항을 추가로 작성한다.

❖ 부족한 점

- 완전성 부족: 시작과 종료 상황이 고려되지 않음

➔ **따라서 요구사항의 추가가 필요함**

일련번호	V5
이름	프로그램 실행
행동	비디오 재생 프로그램은 사용자가 비디오 파일을 더블 클릭하게 되면 자동적으로 실행된다. 프로그램이 실행되면, 가로방향으로 이미지를 보여주는 제어 패널을 담고 있는 윈도우를 화면에 띄운다. 제어 패널은 사용자가 응용 프로그램을 중지할 때 까지 윈도우에 위치한다. 만약 프로그램이 실행중이라면, 다른 비디오 파일을 더블 클릭하는 것은 무시된다.

일련번호	V6
이름	프로그램 정지
행동	비디오 재생 응용 프로그램이 실행 중일 때, 사용자는 비디오가 보여주는 영역을 더블 클릭할 수 있다. 이 때 프로그램은 팝업 윈도우를 띄우는데 이 팝업 윈도우는 **중지**, **계속**의 두 버튼을 가진다. 만약 사용자가 **중지** 버튼을 클릭하게 되면 응용 프로그램은 실행을 중지한다. 만약 사용자가 **계속** 버튼을 클릭하게 되면 비디오 재생은 계속된다.

❖ 논리적 추론의 두 가지 방법론

1. 컴퓨터 소프트웨어의 동작에 대한 기능적 요구사항을 원인-결과 관계로 분석하고 이들 내용을 컴퓨터 명령어로 바꿈
2. 연역적 추리 방법을 응용하여 프로그래밍함

❖ 원인 결과 관계는
- 작업(결과)을 수행하게 하는 논리적인 조건(원인)으로 구성

❖ IF (원인) then (결과) 의 방식으로 프로그래밍
- 예) IF (사용자 이름과 패스워드가 입력됨)
  then (사용자 이름과 패스 워드가 유효한지 확인)

원인	결과
사용자 이름과 패스워드가 입력됨	사용자의 이름과 패스워드가 유효한지 확인
사용자는 윈도우의 닫기 버튼을 클릭	화면에서 윈도우 화면이 사라짐
노트북 컴퓨터 배터리가 10% 미만으로 남음	배터리가 얼마 안남았다는 경고화면을 띄움
인터넷에서 새 이메일이 옴	사용자에게 알람 벨을 울림

그림 4.4 일반적인 프로그램의 원인–결과 관계

❖ 특정 상황에 일반적인 규칙을 적용하여 문제 해결

- 셜록홈즈가 사용하는 추리 방법 (99P)
- 빗변의 길이(c) (특정상황)를 구하기 위해
  피타고라스 정리(일반규칙 $a^2 = b^2 + c^2$) 사용

❖ 문제를 해결하기 위한 작업들의 선후관계를 기술

- 문제를 해결하기 위한 알고리즘을 기술할 때 필요한 작업들을 선후관계에 맞추어 기술해야 함.
- 정확한 순서를 가져야 유효한 알고리즘이 됨

1)
```
priceWithTax ← 0
itemCost ← 100
priceWithTax ← itemCost + itemCost*0.55
```

2)
```
itemCost ← 100
priceWithTax ← itemCost + itemCost*0.55
priceWithTax ← 0
```

- 같은 작업들이라고 해도 서로 다른 순서에 의해 서로 다른 값이 저장됨.
  1) priceWithTax: 105.5
  2) priceWithTax: 0

❖ 문제를 해결하기 위한 규칙들은 '패턴'의 형태로 구성될 때가 있음.

❖ 두 변수의 내용을 서로 교환하는 문제는 교환 패턴을 사용하여 구현할 수 있음.

```
temp ← varA
varA ← varB
varB ← temp
```

그림 4.5 교환 패턴

- myDog와 yourDog 변수의 내용을 교환하는 문제의 해결을 위해 교환 패턴을 사용

```
temp ← myDog
myDog ← yourDog
yourDog ← temp
```

- 특정 작업이 반복적으로 실행될 필요가 있을 때 사용되는 패턴
- 컴퓨터가 가장 잘 하는 것이 지루한 작업을 반복적으로 계속 수행하는 것이며 따라서 컴퓨터 프로그램에서 가장 흔하게 발생함.

❖ **특수한 규칙에서 일반적인 특성을 유도하는 데 사용**

- 귀납 추론을 통해 패턴을 도출함
- cf) 추론-일반적 규칙에서 특수한 특성을 유도하는데 사용

❖ **3단 소프트웨어 구조 패턴**

- 예) 인터넷 서버는 다음의 3단 소프트웨어 구조 패턴으로 구현됨
  1. 사용자와 통신하는 소프트웨어
  2. 데이터를 읽고 저장하는 소프트웨어
  3. 사용자 입력과 저장된 데이터를 기반으로 계산을 하는 소프트웨어

❖ **3단 소프트웨어 구조는 많은 성공적인 소프트웨어 솔루션들의 구조를 관찰함에 의해 특수한 규칙이 파악된 구조**

- 귀납추론의 한 예

❖ **분할정복 (divide and conquer) 이라고도 불림**

❖ **하나의 문제를 여러 부분 문제들로 나누고 그 부분 문제들을 해결함에 의해 원래 문제를 해결하는 방법**
- 복잡한 문제를 그보다는 단순화된 부분 문제들로 분해하여 부분 문제를 해결하고 이를 통해 원래 문제를 해결함

❖ **두 가지 주요 방법론**
- 하향식 설계
- 연속 프로토타이핑

❖ 동영상
- https://www.youtube.com/watch?v=5Ewamx4fRUI
- https://www.youtube.com/watch?v=zTeNgZk7weY

❖ 예) 분할 정복 예 – 피자 만들기 문제
피자 만들기 문제를 6 가지 부분 문제로 분할 후, 6가지 부분 문제를 모두 해결함에 의해 피자 만들기 문제를 해결

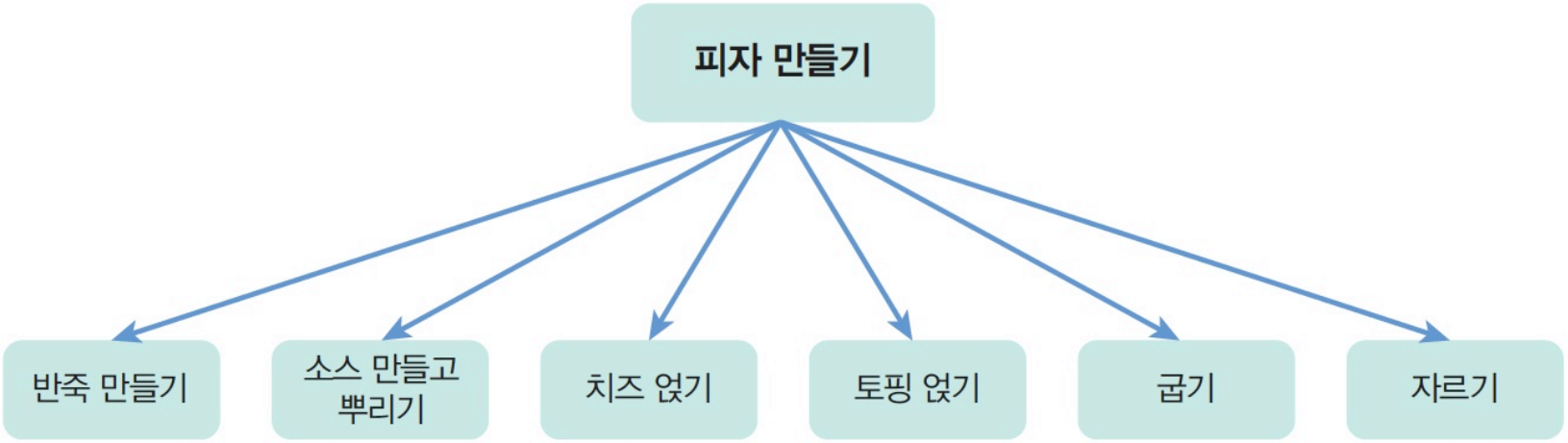


그림 4.7 피자를 만드는 문제를 분해하기

❖ 문제의 요약에서부터 시작

- 개요작성하기

❖ 분명하지 않은 부분들을 다듬어 점차 구체적으로 분명하게 만듦

❖ 개요작성하기

- 기본적으로 전체 작업을 주요 아이디어 위주로 분해 정리
- 주요아이디어는 세부 항목으로 나뉠 수 있음
- 분명하지 않은 부분들을 다듬어 점차 구체적으로 분명하게 만듦

I. 주 아이디어
    A. 아이디어 I의 세부항목
        1. I.A의 세부항목
        2. I.A의 세부항목
    B. 아이디어 I의 세부항목
        1. I.B의 세부항목
        2. I.B의 세부항목
        3. I.B의 세부항목
            a) I.B.3의 세부항목
            b) I.B.3의 세부항목
II. 주 아이디어
    A. 아이디어 II의 세부항목
    B. 아이디어 II의 세부항목
III. 주 아이디어

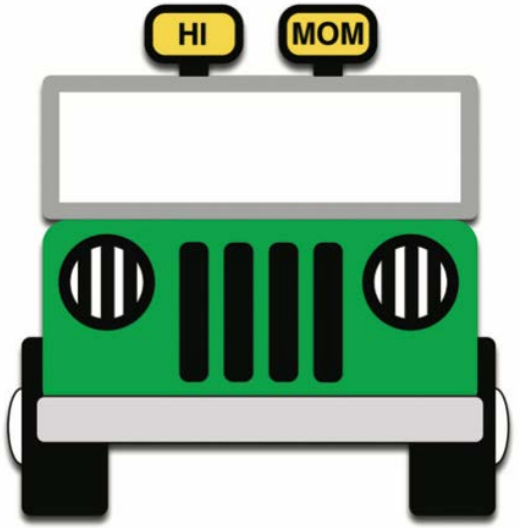
그림 4.8 표준 개요 작성하기 양식

- **설계(design) – 문제정의 단계에서 정리된 소프트웨어 요구사항 명세서에 기술된 내용을 실제로 구현할 수 있는 수반되는 모든 개발과정**

- **알고리즘 - 특정작업을 수행하기 위한 명령어 모음**

- **하양식 설계 – 문제의 요약을 통해 분명하지 않은 명령어들을 명확하게 다듬어 표현하는 방법**
  - 예) 비포장 도로용 차량 그리기 문제

1. 녹색 그릴을 그린다.
2. 그릴 바로 밑에 범퍼를 그린다.
3. 그릴과 범퍼 아래에 타이어를 그린다.
4. 그릴 바로 위에 앞유리를 그린다.
5. 앞유리의 위에 보조등을 두개 그린다.

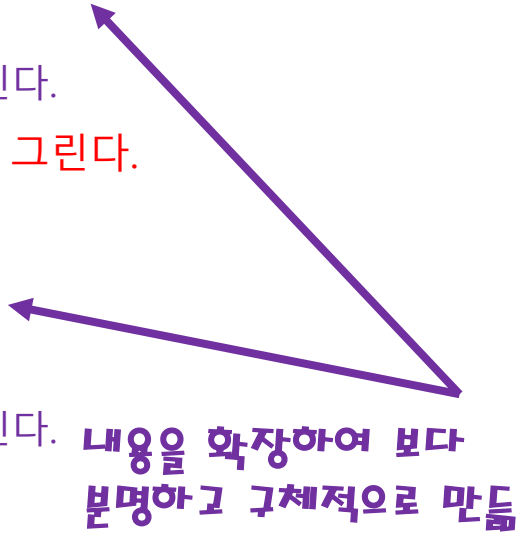


❖ 비포장 도로용 차량: 두번째 단계 알고리즘

1. 녹색 그릴을 그린다.
   (1) 녹색 그릴의 배경을 그린다.
   (2) 좌측 헤드라이트를 그린다.
   (3) 그릴 앞면에 4 개의 동일한 모양과 크기의 검정색 직사각형을 그린다.
   (4) 우측 헤드라이트를 그린다.
2. 그릴 바로 밑에 범퍼를 그린다.
3. 그릴과 범퍼 아래에 타이어를
   (1) 그릴과 범퍼의 좌측부분에서 조금 튀어나오게 좌측 타이어 (검정색 직사각형)를 그린다.
   (2) 좌측 타이어의 바깥쪽에 반달모양의 후드캡을 그린다.
   (3) 그릴과 범퍼의 우측부분에서 조금 튀어나오게 우측 타이어 (검정색 직사각형)를 그린다.
   (4) 우측 타이어의 바깥쪽에 반달모양의 후드캡을 그린다.
4. 그릴 바로 위에 앞유리를 그린다.
   (1) 앞유리창 테두리에 회색 직사각형을 그린다.
   (2) 앞유리창 중앙에 흰색 직사각형을 그린다.
5. 앞유리의 위에 두 개의 보조등을 그린다.
   (1) 좌측 보조등을 그린다.
   (2) 우측 보조등을 그린다.

❖ 비포장 도로용 차량: 최종 단계

1. 녹색 그릴을 그린다
   (1) 녹색 그릴의 배경을 그린다.
   (2) 좌측 헤드라이트를 그린다.
      1) 좌측 헤드라이트의 테두리로서 검은 색 점선을 그린다.
      2) 검은 색 테두리 안에 흰색 점을 가운데 그린다.
      3) 헤드라이트 보호개용으로 세 개의 같은 모양의 직사각형을 그린다.
   (3) 그릴 앞면에 네 개의 동일한 모양과 크기의 검정색 직사각형을 그린다.
   (4) 우측 헤드라이트를 그린다.
      1) 좌측 헤드라이트의 테두리로서 검은 색 점선을 그린다.
      2) 검은 색 테두리 안에 흰색 점을 가운데 그린다.
      3) 헤드라이트 보호개용으로 세 개의 같은 모양의 직사각형을 그린다.



내용을 확장하여 보다 분명하고 구체적으로 만듦

❖ 프로토타입 (prototype)

- 어떤 물체의 대략적인 모습
- 최종 제품을 만들기 전에, 일부분 만을 보여 주어 최종 제품의 모습을 대략 예상할 수 있도록 함

예) 자동차 프로토타입, 아파트 모델 하우스

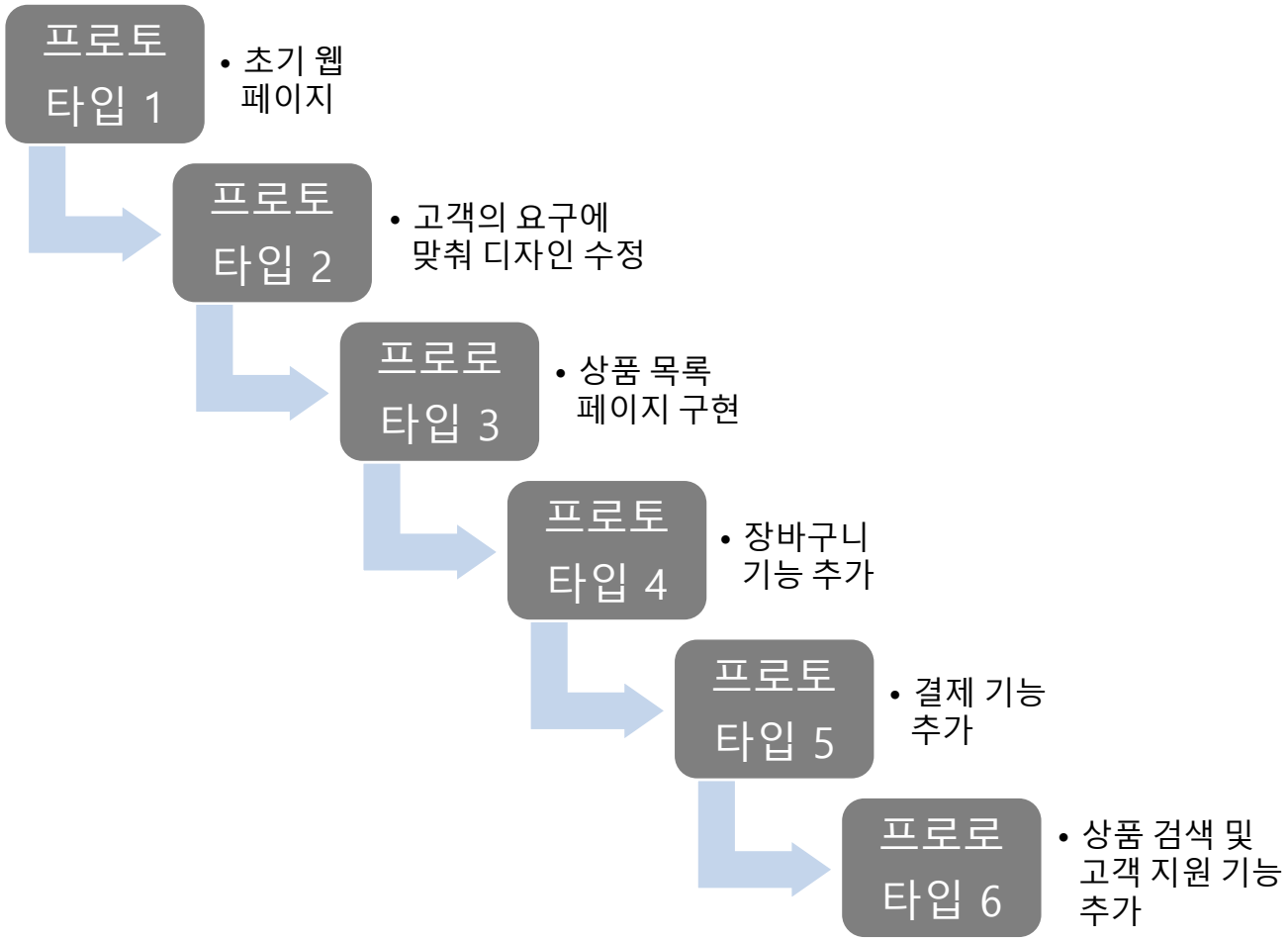
❖ 연속 프로토타이핑 (successive prototyping)

- 프로토타입을 설계 단계 별로 계속 만들어 설계 단계 마다 제품의 최종 모습을 미리 어느 정도 파악할 수 있도록 함
- 각 단계에서 완성된 프로토타입은 고객에게 보여지고 피드백 받음

❖ 연속 프로토타이핑의 예 : Sal이라는 사람의 서핑보드 가게 (SSS)

그림 4.12 SSS 온라인 쇼핑몰의 초기 웹 페이지

4.3 문제분해


프로토 타입 1
• 초기 웹 페이지
프로토 타입 2
• 고객의 요구에 맞춰 디자인 수정
프로로 타입 3
• 상품 목록 페이지 구현
프로토 타입 4
• 장바구니 기능 추가
프로토 타입 5
• 결제 기능 추가
프로로 타입 6
• 상품 검색 및 고객 지원 기능 추가

❖ 멀티태스킹 (multitasking)

- 하나의 소프트웨어에서 여러 부분을 동시에 실행
- 하나의 코어가 여러 작업을 동시에 수행함
  예) 수퍼컴퓨터 – 수만 개의 코어

❖ 그리드컴퓨팅 (grid computing)

- 하나의 작업을 여러 코어가 작업을 나누어 동시에 수행함

❖ 탐색 알고리즘

- 선형 탐색: 순서대로 나열된 데이터를 순서대로 원하는 데이터를 찾을때 까지 탐색
- 이진 탐색: 목록에서 중간값을 사용, 특정 이름의 내용 찾기
  - 선형탐색보다 빠름

❖ **문제**

1000개의 가나다 순서로 배치된 성적표 더미에서 당신의 성적표 찾기

❖ **이진 탐색 방법**

1. 남은 더미의 중간 성적표 확인
2. 중간 성적표가 당신의 것이면 종료
3. 중간 성적표의 이름이 가나다 순으로 당신 이름 보다 뒤라면 뒷부분 더미를 탐색에서 제외 그렇지 않으면 앞 부분 더미를 탐색에서 제외
4. 단계 1 – 3 반복

❖ **선형탐색과 이진 탐색에서 단계가 수행된 횟수는?**

4.4 분해: 다른 사용법들

선형탐색	
탐색 단계 수	탐색을 해야하는 남은 자료 숫자
1	499
2	498
3	497
4	496
5	495
6	494
7	493
8	492
9	491
10	490
...	...
499	1
500	탐색 완료

이진탐색	
탐색 단계 수	탐색을 해야하는 남은 자료 숫자
1	250
2	125
3	62
4	31
5	15
6	7
7	3
8	1
9	탐색 완료

그림 4.13 선형탐색과 이진탐색의 최악의 경우의 성능(500개 자료 기준)

❖ **복잡한 문제를 추상화 (abstraction) 를 사용 단순화 시킨 후 문제를 해결함**

- 문제의 가장 중요한 이슈들에게만 집중함
- 제어의 추상화, 자료의 추상화, 행위의 추상화를 통해 문제를 해결할 수 있음
- 동영상 (2m28s), https://www.youtube.com/watch?v=mZkuvP_PsDI

❖ **제어구조 (control structure)**

- 명령어가 수행되는 순서를 서술하는 체제
- 제어의 추상화는 제어구조의 형태로 표현함

- 알고리즘의 5가지 기본 제어구조
  1. 순차제어
  2. 선택
  3. 반복
  4. 제어의 추사화
  5. 병렬처리 (11장 주제)

  * 알고리즘의 제어흐름은 다섯 가지 구조들을 혼용하여 구성

❖ 퍼지 요리법은 순차실행의 많은 예를 보여준다.

요리법

1. 다음 재료들을 전자레인지에 사용가능한 접시에 넣는다:
   초코렛 칩 3 컵
   압축 우유 14 온스 1컵
   버터 ¼ 컵

선택

2. 원한다면, 1 컵의 호두 조각을 넣어 젓는다.
3. 1분간 전자레인지에서 익힌다.
4. 전자레인지에서 혼합물을 꺼낸 후 다시 젓는다.

반복

5. 초코렛 칩이 완전하게 녹을 때 까지 3단계와 4단계를 반복한다.
6. 바닐라 1 스푼을 혼합물에 넣는다.
7. 혼합물을 8 × 8 크기의 접시에 넣는다.
8. 3 시간동안 냉장시킨다.
9. 퍼지를 1인치 크기의 정사각형으로 자른다.

그림 4.14 퍼지 요리법

❖ **알고리즘의 한 명령어가 또 다른 부분 알고리즘의 실행을 참조할 때 발생**

- 추상화를 통해 산만하게 보일 수 있는 상세 내용을 제거함

요리법

1. 퍼지를 만든다(그림 4.14 참조).
2. 초코렛 칩 쿠키를 만든다.
3. 땅콩 버터 바를 만든다.
4. 퍼지, 쿠키, 바 들을 큰 쟁반에 배치한다.

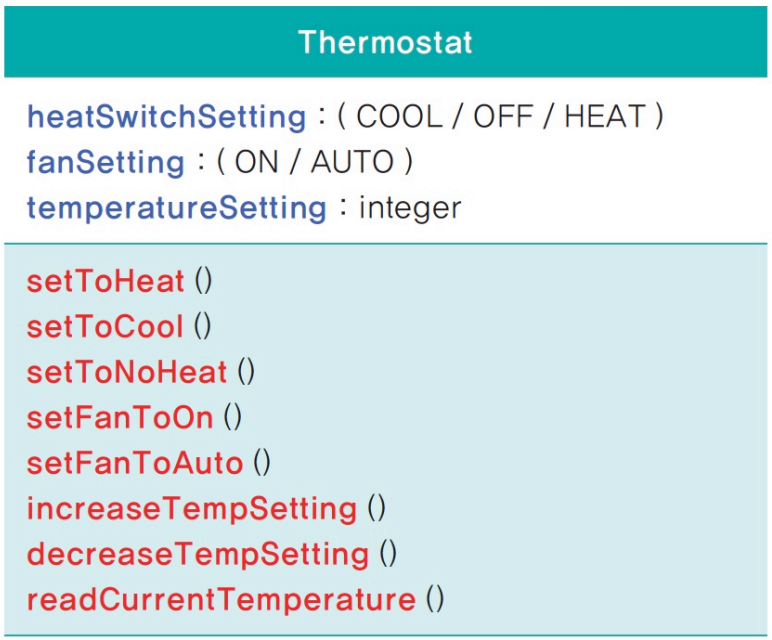
그림 4.15 간식 세트 만들기

❖ **클래스 다이어그램 (class diagram): 데이터 추상화를 그림으로 나타내는 표준**
- 속성(attributes)와 동작 (operations)에 대한 객체의 추상화
- 예) 자동온도조절기 (Thermostat)의 클래스 다이어그램

클래스이름

속성

동작



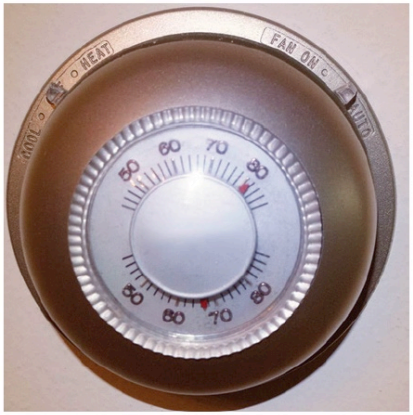


그림 4.16 Thermostat 클래스 다이어그램

* 동작의 각 ( )는 매개변수가 포함될 수 도 있음

- 매개변수를 통해 전체 동작들의 목록을 간소화 할 수 있다.

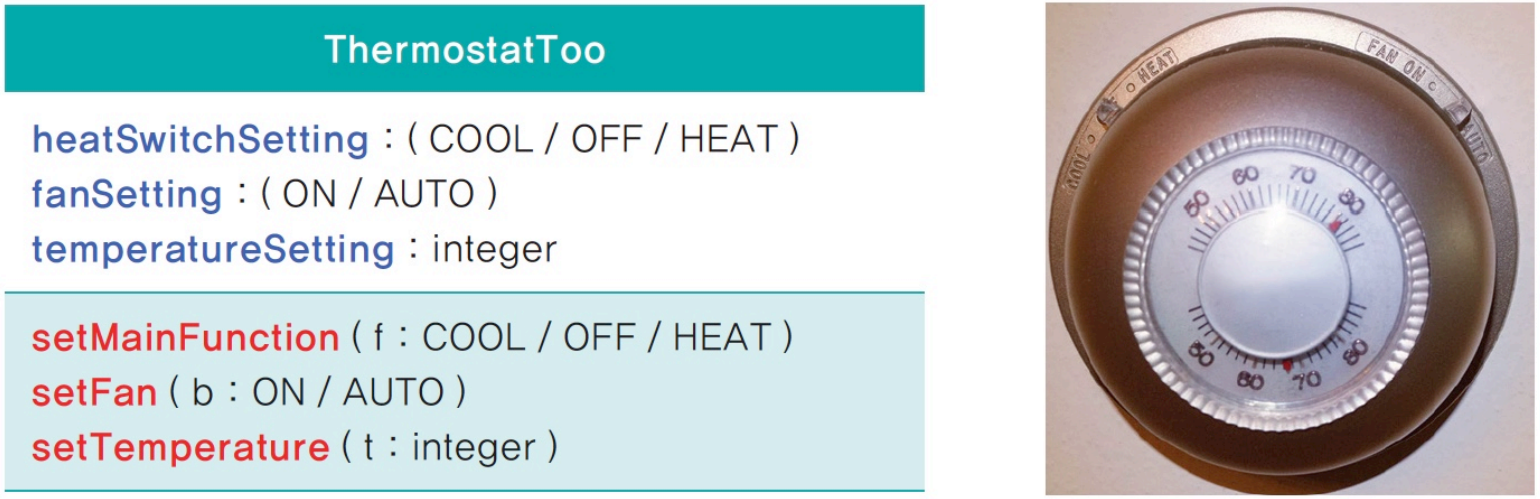


그림 4.17 Thermostat Too 클래스 다이어그램

❖ 쓰임새 다이어그램 (유스케이스 다이어그램, 사용 시나리오)

- 어떤 시스템을 컴퓨터 사용자와 시스템간의 관계로 표현하는 것
- 중요요소
  - 행위자(Actor): 같은 종류의 사용자 집단
  - 쓰임세 (use case): 시스템에 의해 행해질 수 있는 행동
  - 행위자와 그 행위자가 할 수 있는 특정한 행동들간에는 선으로 연결
  - 쓰임세의 행위자들은 역할(role)로 분류됨

    예) 온라인시스템 – 학생, 담당직원, 교수

4.6 추상화: 쓰임새 다이어그램

❖ **액터**
- 학생, 담당직원, 교수

❖ **유스케이스**
- 과목등록
- 과목등록 취소
- 로긴
- 과목 보기
- ...

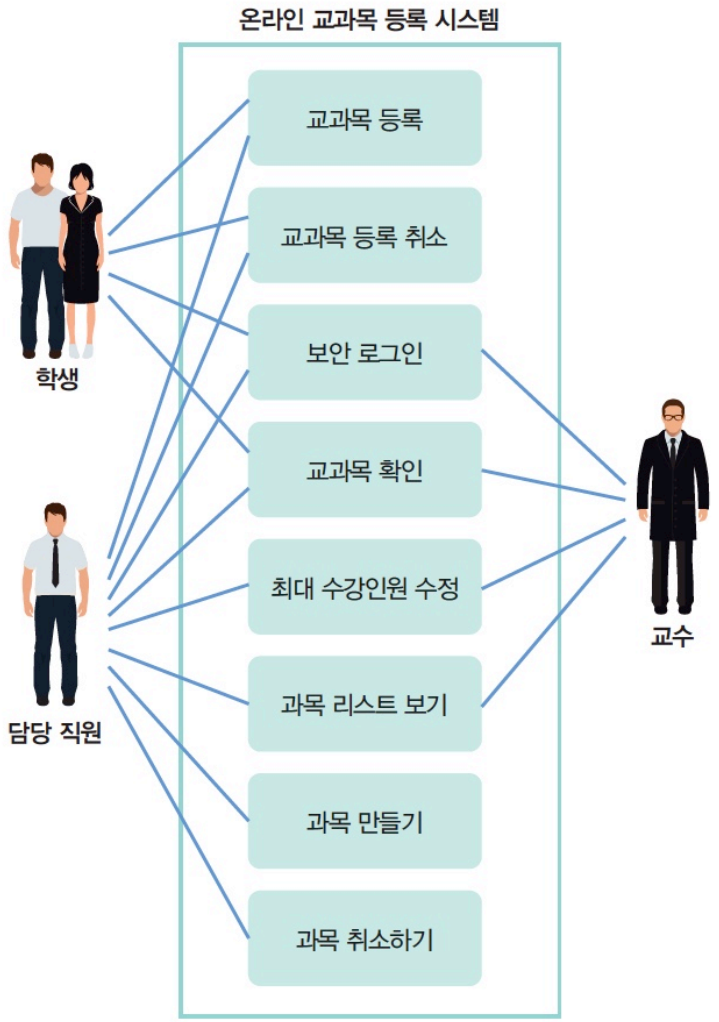


그림 4.20 온라인 교과목 등록 시스템 유스 케이스

4.6 추상화: 쓰임새 다이어그램

❖ 액터
- 고객
- 점원

❖ 유스케이스
- 물품구입
- 술 구입
- 나이 확인
- …

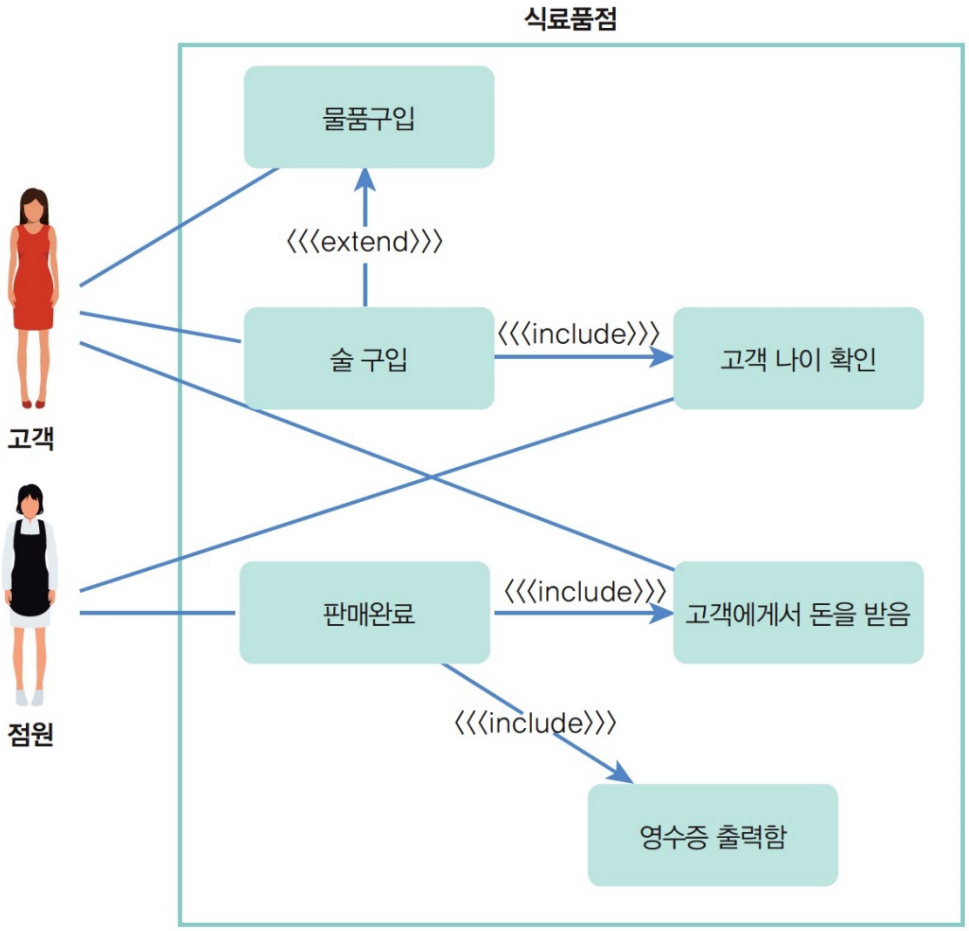


그림 4.21 식료품점 유스 케이스 다이어그램

❖ 행위자들간의 관계 표현
- <<extended>>
  어떤 행동이 다른 행동의 확장된 행동이거나 특수화된 행동인 경우
- <<included>>
  하나의 행동이 그 기능의 일부로서 다른 행동을 수반하는 경우

❖ **컴퓨팅사고의 핵심내용**

- 문제정의, 논리적 추론, 분해, 추상화

❖ **제어패턴**

- 순차진행, 선택, 반복, 제어 추상화, 병렬처리
- 알고리즘의 형태로 표현됨

❖ **다이어그램**

- 상태활동 다이어그램
- 유스케이스 다이어그램
- 클래스 다이어그램

4.7 참고문헌

- 문봉교, 김웅섭 역, 컴퓨팅 사고 (소프트웨어를 통한 문제해결), 인피니티북스, 2017

- 발표방식: ppt 자료 활용 2개 조 각각 10분 발표 (질의응답 포함)

- 문제 1: A그룹
  - 최근 출시된 카카오택시 어플리케이션은 택시를 호출하는 사용자들에게 편의성을 제공해 주는 어플리케이션이다. 카카오택시 어플리케이션의 다양한 기능과 이러한 기능을 이용하는 고객의 관계를 유스케이스 다이어그램을 작성하시오.

- 문제 2: B그룹
  - 러닝머신은 사람이 앞으로 뛰며 이동하는 상식을 뒤집은 역 발상을 통한 문제 해결 방법이다. 이러한 역 발상을 통한 문제 해결방식의 다양한 사례를 제시하시오. 그러한 문제해결방식이 어떤 측면에서 역발상이라고 할 수 있는지 논의해 보시오.