

20장

오래 사용할 수 있는 프로그램 만들기

20장 오래 사용할 수 있는 프로그램 만들기

- 20.1 큰 작업을 더 작은 작업들로 나누기
- 20.2 프로그래밍에 블랙박스 도입하기
- 20.3 자신만의 환경에서 존재하는 하위 작업들
- 20.4 요약

20.1 큰 작업을 더 작은 작업들로 나누기



20.1.1 온라인으로 상품 주문하기

» 온라인에서 물건을 주문하면 발생하는 과정

1. 주문을 넣기 위한 웹 폼을 작성한다. 주문 정보를 판매자에게 전송하면 판매자는 품목, 수량, 주문자 이름, 주소 등의 중요한 정보를 뽑아낸다.
2. 상품 분류와 번호에 따라서 판매자(사람 또는 로봇)는 창고에서 상품을 찾아 포장 담당자에게 넘긴다.
3. 포장 담당자는 박스에 상품(들)을 넣는다.
4. 주문자의 이름과 주소로 송장을 작성한다.
5. 박스에 송장을 붙이고 우체국에 박스를 보낸다. 우체국은 주문자의 주소를 찾아서 상품을 배달한다.



20.1.1 온라인으로 상품 주문하기

» 작업 의존성/독립성 (task dependence/independence)

- 어떤 작업이 끝나야 다른 작업을 시작할 수 있는 경우, 후자가 전자에 의존적(dependent)이라고 말한다.
- 두 작업이 서로 상관 없이 아무 때나 동시에 진행될 수 있으면 두 작업은 서로 독립적(independent)이라고 말한다.

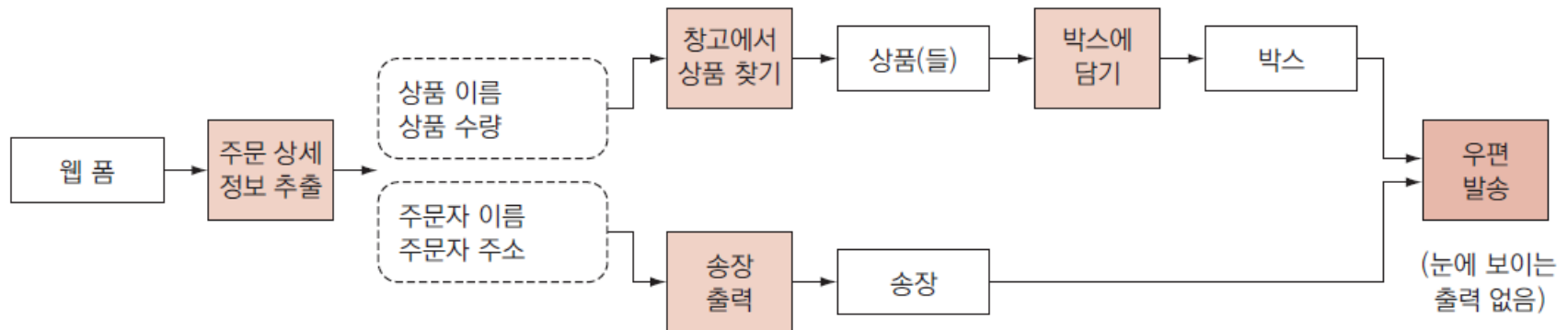


그림 20-1

온라인에서 상품을 주문하는 작업을 더 작고, 자기 완결성이 있으며, 재사용 가능한 하위 작업으로 나누는 방법. 컬러 상자는 한 가지 작업을 표시한다. 컬러 상자 왼쪽에 있는 것은 그 작업에 대한 입력이고, 오른쪽에 있는 것은 그 작업의 출력이다

» 셀프 체크 20.1



20.1.1 온라인으로 상품 주문하기

» 추상화(abstraction)

- 작업을 추상화(abstraction)한다는 말은 작업을 단순화하는 방법 중 하나로, 작업을 이해하는데 필요하지 않은 상세 사항을 모두 감춰 최소한의 정보만 사용해 작업을 이해할 수 있게 한다.
- ‘창고에서 상품찾기’ 경우 ‘상품 이름’과 ‘상품 수량’만 입력하면 요청한 상품을 받을 수 있다는 사실을 아는 것만으로 충분하다.

» 재사용 가능한 하위 작업(reusable subtask)

- 다른 입력마다 다른 출력을 만들어낼 수 있어서 여러 곳에서 어떤 하위 작업을 사용할 수 있다면, 그 하위 작업을 재사용 가능(reusable)하다고 말한다.



20.1.2 핵심 이해하기

» 작업을 다룰 때는 각 작업을 블랙박스(black box)로 생각하라.

- 블랙박스는 어떤 작업을 수행하는 시스템을 시각화하는 방법 중 하나다. 시스템 전체 내부를 들여다 볼 수 없는 블랙박스처럼 생각하는 것은, 그 시스템이 하는 일을 이해하기 위해 꼭 그 시스템의 내부를 들여보아야 하는 건 아니라는 점을 상기시켜준다.

» 각 작업이 목표를 어떻게 달성하는지 그 방법을 알 필요가 없다. 하위 작업이 제공하는 기능만 가지고 전체 시스템을 이해하려고 노력하면 된다.

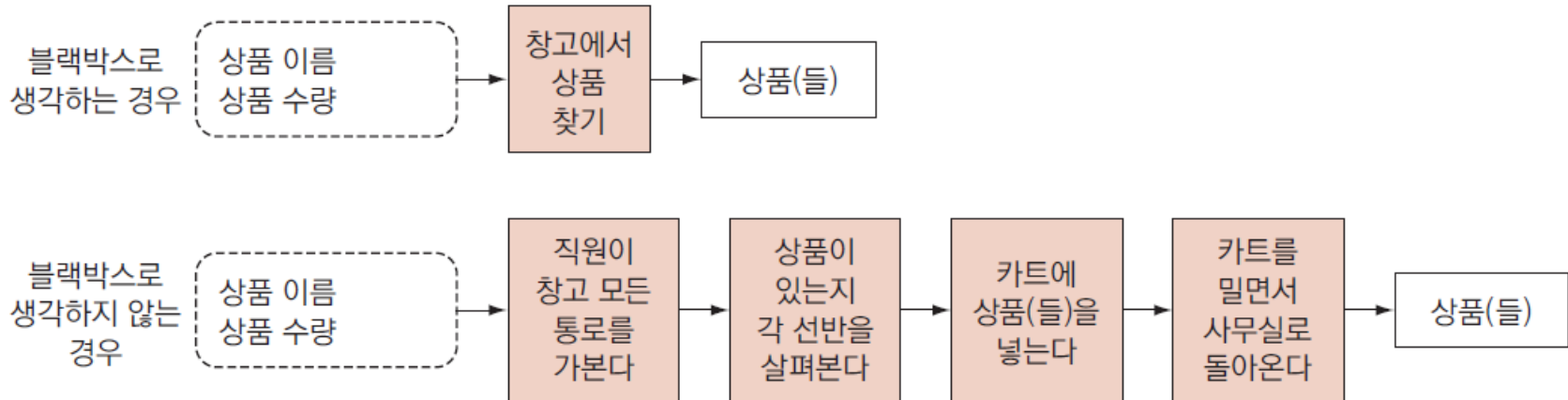


그림 20-2

‘창고에서 상품 찾기’를 블랙박스처럼 살펴보는 경우와 그렇지 않은 경우를 비교한 그림. 창고에서 상품을 찾는 자세한 과정을 본다고 해서 ‘창고에서 상품 찾기’라는 작업 자체에 대한 이해가 더 깊어지지는 않는다

20.2 프로그래밍에 블랙박스 도입하기



20.2.1 코드 모듈 사용하기

- » 코드 모듈(code module)은 어떤 작업을 하기 위한 코드 조각을 말한다. 모듈마다 각각 입력, 처리, 출력과 연관된다.
- » 모듈성(modularity)은 큰 프로그램을 작은 작업들로 얼마나 잘 나누었느냐를 말한다.
 - 모듈화를 잘하면 각 하위 작업의 코드를 서로 독립적으로 따로 작성할 수 있다.
 - 일반적으로 각 코드 모듈은 그 자체로 완결성이 있어야 한다. 또 모듈이 제대로 작동하는지 여부를 쉽게 테스트할 수 있어야 한다.
 - 이런 식으로 큰 작업을 나누면 큰 문제가 더 쉬워 보이고, 디버깅에 드는 시간을 줄일 수 있다



20.2.2 코드 추상화하기

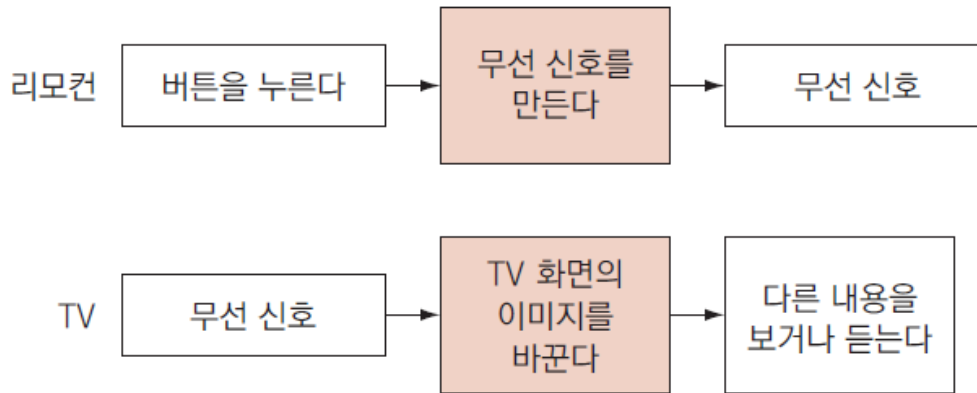


그림 20-3
리모컨과 TV를 블랙박스로 본 경우

아이템	입력	처리	출력
리모컨	버튼을 누른다	눌린 버튼에 따라 다른 신호를 만든다	무선 신호
TV	리모컨이 보낸 무선 신호	화면의 이미지가 바뀌거나(전체 이미지가 바뀌거나 볼륨 표시 막대가 표시됨) 볼륨이 바뀐다	사용자가 보거나 듣는 화면이나 소리 변화

표 20-1
채널을 바꾸거나 볼륨을 바꿀 때 TV와 리모컨의 입력, 처리, 출력



20.2.2 코드 추상화하기

- » 프로그래밍에서 추상화를 사용하는 이유는 아이디어를 더 높은 수준에서 표현하기 위해서다.
- » 추상화는 코드 조각이 하는 일을 세 가지 핵심 정보인 입력, 처리, 출력을 가지고 문서화하는 과정이다.
- » 모듈을 문서화할 때는 특별한 주석을 사용하며, 이를 독스트링(docstring)이라고 한다. 독스트링에는 다음과 같은 정보가 들어간다.
 - 모듈에 들어오는 모든 입력 : 변수와 변수의 타입을 명시해서 표시한다.
 - 모듈이 해야 하는 처리 : 모듈의 기능을 말이나 수식 등으로 설명한다.
 - 모듈이 제공할 출력 : 모듈이 만들어내는 객체(변수)나 모듈이 콘솔에 내보내는 출력 등을 설명한다.



20.2.3 코드 재사용

» 동일한 네 가지 연산을 여러 다른 수에 적용하는 이런 상황이 코드 재사용이 유용해진다.

```
a = 1
b = 2
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```

----- a = 1, b = 2일 때 연산을 수행하는 코드

```
a = 3
b = 4
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```

----- a = 3, b = 4일 때 연산을 수행하는 코드

```
a = 5
b = 6
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```

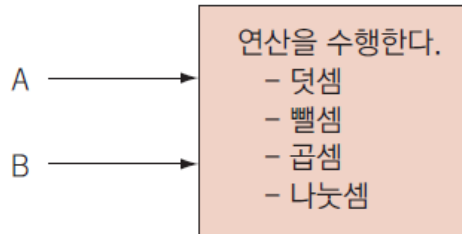
----- a = 5, b = 6일 때 연산을 수행하는 코드

코드 20-2
쌍의 숫자에 대해 덧셈, 뺄셈,
곱셈, 나눗셈 수행하기



20.2.3 코드 재사용

- 공통 과정을 블랙박스로 만들고, 블랙박스의 입력을 바꿔라(그렇게 하면 출력은 저절로 바뀔 것이다).



- 이 박스는 코드를 둘러싸는 코드 래퍼(wrapper)로, 프로그램에 기능을 추가한다. 이런 래퍼를 함수(function)라 부른다.
- 변수 a와 b의 값을 여전히 바꿔야 하지만, 이제는 코드 래퍼를 통해 블랙박스를 사용할 수 있다.
- 네 가지 산술 연산을 처리하는 네 줄짜리 코드를 블랙박스를 사용해 한 묶음으로 단순화했다.

```
a = 1
```

```
b = 2
```

```
<a와 b에 대한 네 가지 연산을 감싼 래퍼>
```

```
a = 3
```

```
b = 4
```

```
<a와 b에 대한 네 가지 연산을 감싼 래퍼>
```

```
a = 5
```

```
b = 6
```

```
<a와 b에 대한 네 가지 연산을 감싼 래퍼>
```

(진짜 코드가 아님) 네 가지 연산을
수행하는 블랙박스의 위치를 표시

코드 20-3
세 쌍의 수를 더하고, 빼고,
곱하고, 나누기

20.3 자신만의 환경에서 존재하는 하위 작업들



20.3 자신만의 환경에서 존재하는 하위 작업들

- » 세 사람이 전화의 역사를 연구해서 발표해야 하는 그룹 프로젝트를 진행한다고 하자.
- » 코드 모듈을 어떤 목표를 달성하기 위해 실행되는 작은 프로그램이라고 생각할 수 있다.
- » 각 모듈은 자신만의 환경에서 실행되며, 한 모듈의 환경은 다른 모듈의 환경과 서로 독립적이다.
- » 어떤 모듈이 만든 객체는 그것을 명시적으로 다른 모듈에 넘기지 않는 한, 그 객체를 만들어낸 모듈에만 고유하게 존재한다.
- » 모듈은 입력과 출력을 통해 객체를 넘길 수 있다.

20.4 요약



20.4 요약

- » 모듈은 서로 독립적이며, 자신만의 환경에서 실행된다.
- » 코드 모듈을 한 번만 작성해도 다른 입력을 가지고 여러 번 재사용할 수 있다.
- » 모듈의 세부 사항을 보지 않아도 되도록 추상화하면 많은 모듈이 더 큰 작업을 달성하기 위해 서로 협력하는 방식에 더 집중할 수 있다.



20.4 요약

» (Q20.1) ‘커플이 식당에 가서 음식과 음료를 시킨다’라는 작업을 더 작은 하위 작업으로 나눠보라. 입력과 출력, 이 작업과 관련한 여러 사람들 사이의 상호 관계가 들어가는 블랙박스 다이어그램을 그려 보라.