

일 반 적 으 로 권 고 한 객 체 지 향 설 계 는 다음 단계가 포함

1. 적 절 한 분 석 수 행

2. 시 스템 을 설 명 작 업 명 세 서 개 발

3. 이 작 업 명 세 서 로 부터 모 구 사 항 을 수 집

4. 사 용 자 인 터 페이스 용 프 로 토 타입 개 발

5. 클 래 스 식 별

6. 각 클 래 스 의 역 할 을 결 정

7. 다 양 한 클 래 스 가 서 로 상 호 작 용 하 는 방 식 결 정

8. 만 들 고 자 하 는 시 스템 을 설 명 하 는  
고 급 모 델 구 성

## 객체 래퍼

객체 지향과 구조적 프로그래밍은 함께 사용됨

구조적 코드는 순서, 조건, 반복으로 이루어짐

↓            if            for

구조적 설계에서는 코드를 함수로 둘러싸고

객체 지향 설계에서는 코드를 객체와 메서드로 둘러싼다.

객체 래퍼를 이용하면 네이티브 코드를 감출 수 있다.

기본 클래스를 객체 래퍼로 둘러싸 기본 클래스를

새로운 클래스 안에 둘러싸서 기본 클래스의 구현이나  
인터페이스를 바꿀 수 있다.

## 결론

이번 장에서는 전체 시스템의 설계 과정을 설명했다. 객체지향 코드와 구조적 코드는 서로 배타적인 게 아니다. 실제로 여러분은 구조적 코드를 사용하지 않으면 객체를 만들 수 없다. 따라서 여러분이 객체지향 시스템을 구축하는 과정이라면 이미 설계 시에 구조적 기술을 사용하는 셈이나 마찬가지다.

객체 래퍼는 전통적인 구조적 코드(레거시)와 객체지향적인 코드(클래스)에서 이식 불능 코드(네이티브)에 이르기까지 다양한 기능을 캡슐화하는 데 사용된다. 객체 래퍼의 주요 목적은 코드를 사용하는 프로그래머에게 일관된 인터페이스를 제공하는 데 있다.

다음 몇 장에서는 클래스 간의 관계를 더욱 자세히 살펴본다. 7장 '상속과 합성에 익숙해지기'에서는 상속 및 합성의 개념과 이들이 서로 관련되는 방법을 설명한다.

