CODE COMPLETE 스터디

6.1 클래스 기초 : 추상 데이터 형(ADT)

컨텐츠검색플랫폼팀
조영일

관점의 변화

- 프로그래머들의 관점
  - 세월에 따른 관점의 변화
  - Statement 〉〉 Routine 〉〉 Class

클래스

- 정의
  - 응집력 있고 잘 정의된 기능을 공유하는 데이터와 루틴의 모음

추상 데이터 형

- ADT(Abstraction Data Type)
  - 데이터와 데이터를 다루는 연산의 집합
  - 예)
    - 그래픽 윈도우
    - 파일과 파일 연산
    - 보험률 표와 연산

주의 사항

- ADT는 그저 컨테이너에 불과한가?
  - No!
- ADT는 저수준 구현 개체인가?
  - No!
- ADT는 실세계와 맞닿아 있는 개념 개체임
  - 예)
    - 연결 리스트에 노드를 추가하라
    - 스프레드시트에 셀을 추가하라

ADT가 필요한 예

- 폰트 렌더러
  - 글꼴, 크기, 굵기, 기울임을 사용하여 화면에 출력되는 텍스트를 제어하는 프로그램 개발

ADT가 필요한 예

- 크기 지정
  - currentFont.size = 16
  - currentFont.size = PointsToPixels(12)
  - currentFont.sizeInPixels = PointsToPixels(12)
- 굵기 지정
  - currentFont.attribute = currentFont.attribute or 0x02
  - currentFont.attribute = currentFont.attribute or BOLD
  - currentFont.bold = True
  - currentFont.setBoldOn();

ADT의 혜택

- 세부적인 구현 사항을 감출 수 있다
- 변경이 전체 프로그램에 영향을 미치지 않는다
- 인터페이스가 보다 많은 정보를 제공하도록 만들 수 있다
- 성능을 향상시키기가 쉽다
- 외관상으로 프로그램이 정확하다는 것을 더 잘 알 수 있다
- 프로그램이 보다 더 스스로를 설명하게 된다
- 프로그램에 모든 데이터를 넘길 필요가 없다
- 저수준 구현 구조체 대신 실세계의 개체들을 다룰 수 있다

더 많은 예

- 생략

가이드라인

- 전형적인 저수준 데이터 형을 저수준 데이터 형이 아닌 ADT로 만들거나 사용하라
- 파일과 같은 일반적인 객체를 ADT로 취급하라
- 간단한 항목이라도 ADT로 취급하라
  - self-descriptive code
- ADT가 저장되어 있는 매체에 독립적으로 ADT를 참고하라
  - 파일이든 메모리든 난 모르겠고...

비 객체지향 환경에서

- C언어와 같은 비 객체지향 환경에서는 어떻게 하지?
  - 클래스 개념 없음
  - 굳이 루틴으로 추상화한다면
    - SetCurrentFontSize(sizeInPoints)
    - SetCurrentFontBoldOn()
    - SetCurrentFontTypeFace(faceName)
  - 여러 글꼴을 동시에 다루려면
    - CreateFont(fontId)
    - DeleteFont(fontId)
    - SetCurrentFont(fontId)

비 객체지향 환경에서

- 선택사항
  - 1. 암시적인 인스턴스를 사용하든가
    - SetCurrentFont(fontId)
    - 각별한 주의가 필요함
  - 2. 명시적으로 인스턴스를 식별하든가
    - font ID 파라미터를 함수마다 추가
  - 3. 데이터를 명시적으로 제공하든가
    - Font 데이터 타입을 만들어서 파라미터로 전달
    - font ID가 필요 없도록