추상 자료형과 자료구조

주요 내용

- 자료구조
- 코드 추상화
- 추상 자료형, 추상 메서드

알고리즘과 자료구조

- 알고리즘: 주어진 문제의 모든 사례에 대해 동일하게 작동하는 단계별 명령의 목록으로 구성된 컴퓨터 프로그램
- 계산 가능한 문제: 문제를 해결하는 알고리즘이 존재할 경우
- 문제 하나에 대해 다양한 알고리즘이 존재 가능
- 또한 다양한 프로그래밍언어로만 구현 가능
- 사용되는 알고리즘과 프로그래밍언어의 특성에 따라 성능이 달라질 수 있음
- 주어진 문제를 해결하는 알고리즘에 필요한 두 요소
    - 자료구조: 문제의 사례 표현
    - 명령문: 문제 해결에 필요한 알고리즘 작성

자료구조

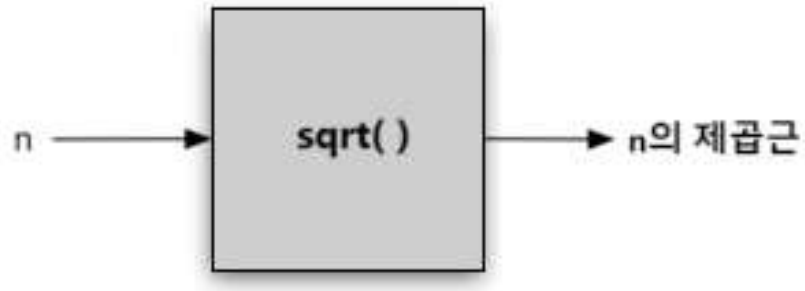
- 프로그램 내에서 데이터를 어떻게 구성하고, 저장하고, 다룰 수 있는지 정의
- 자료구조를 잘 이해하면 자료를 효과적으로 활용하는 알고리즘을 개발할 수 있음
- 대부분의 프로그래밍언어는 정수, 부동소수점, 부울값 등 기본 자료형 이외에 아래 데이터들에 대한 자료구조를 제공
    - 리스트/어레이
    - 튜플
    - 문자열
- 알고리즘 작성에 필수적인 명령문
    - 변수 할당
    - 조건문: `if...else...` 등
    - 반복문: `for`, `while` 등

코드 추상화

- 프로그래밍에서는 많은 종류의 코드 추상화를 사용
- 여기서는 절차 추상화와 데이터 추상화 두 종류만 언급

절차 추상화

- 계산 가능한 문제를 해결하는 알고리즘은 적절한 API(Application Programming Interface)를 조합하여 작성
- 하지만 각각의 API를 어떻게 구현하는가는 사용자 입장에서 관심대상이 아님
- 예제: `sqrt()` 함수가 어떻게 정의되었는가는 모른 채 제곱근을 계산하는 함수라는 사실과 어떻게 활용하는가를 알기만 하면 됨.

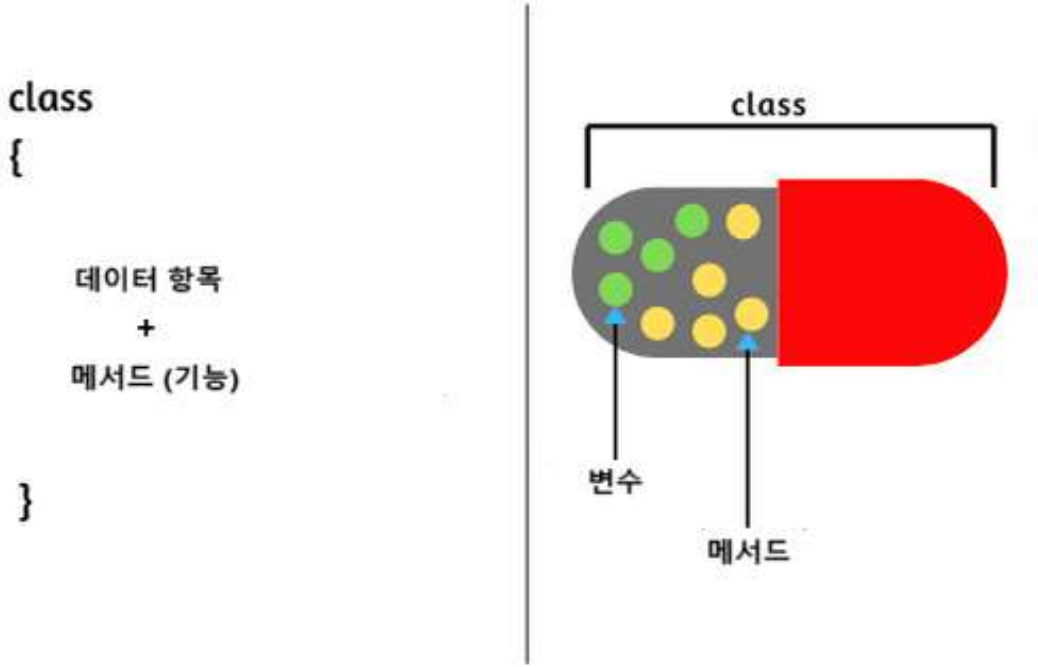


데이터 추상화와 추상 자료형

- **추상 자료형**Abstract data type(ADT): 특정 종류의 데이터와 그 데이터를 다루는 메서드를 하나로 묶어 다루는 방법을 묘사
- **데이터 추상화**data abstraction: 추상 자료형을 이용하여 적절한 자료구조를 정의하는 기법

데이터 캡슐화

- 데이터와 함께 관련 메서드를 하나로 묶어 처리하도록 하는 과정
- 데이터 추상화를 가리키는 다른 표현



추상 자료형 구현

- 클래스 활용
- 사용자의 입장에서 클래스로 선언된 자료구조가 구체적으로 어떻게 구현되었는가는 중요하지 않음.
- 대신에 해당 자료구조가 담고 있는 데이터가 무엇인가와 데이터를 다루는 메서드의 기능을 어떻게 활용할 수 있는가만 알면 됨.

추상 클래스, 추상 메서드, 자료구조

- 추상 클래스abstract class: 추상 자료형에서 명시된 값의 저장 방식과 저장된 값들을 처리하는 필수 기능을 담당하는 메서드 일부를 구체적으로 구현하지 않은 채로 선언하는 클래스
- 추상 메서드abstract method: 본문이 구현되지 않은채로 선언되는 메서드
- 자료구조data structure: 추상 클래스를 상속하면서 동시에 구현되지 않은 메서드를 모두 구체적으로 구현한 클래스