

# Chaper 2. 의미 있는 이름

☑ Reviewed	<input type="checkbox"/>
🕒 생성 일시	@2024년 9월 11일 오후 1:33

- 의도를 분명히 밝혀라

before

```
public List<int[]> getThem() {  
    List<int []> list1=new ArrayList<int[]>();  
    for (int [] x: theList)  
        if (x[0]==4)  
            list1.add(x);  
    return list1;  
}
```

after

```
public List<Cell> getFlaggedCells() {  
    List<Cell> flaggedCells=new ArrayList<Cell>();  
    for (Cell cell: gameBoard)  
        if (cell.isFlagged())  
            flaggedCells.add(cell);  
    return flaggedCells;  
}
```

- 그릇된 정보를 피하라
  - 혼동 될 수 있는 단어 사용 피하기
  - 서로 흡사한 이름 사용하지 않기
  - 유사한 개념은 유사한 표기법을 사용하기
  - 소문자 L, 소문자 O가 발생시킬 수 있는 혼동 주의
- 의미 있게 구분하라
  - 이름이 달라야 한다면 의미도 달라져야 한다.
  - 읽는 사람이 차이를 알도록 이름을 지어라.
- 발음하기 쉬운 이름을 사용하라
  - 코드에 대한 이야기를 나눌 때 지적인 대화가 가능해진다.
- 검색하기 쉬운 이름을 사용하라
  - 문자 하나를 사용하는 이름과 상수는 텍스트 코드에서 쉽게 눈에 띄지 않는다는 문제점이 있다.
    - 간단한 메서드에서 로컬 변수만 한 문자를 사용한다.
    - 이름 길이는 범위 크기에 비례해야한다.
- 인코딩을 피하라
  - 헝가리식 표기법
    - 과거와 달리 헝가리식 표기법이나 기타 인코딩 방식이 오히려 방해가 되며, 독자를 오도할 수 있다.
  - 멤버 변수 접두어
    - 멤버 변수에 더 이상 m\_이란 접두어를 붙일 필요가 없다.
    - 클래스와 함수는 접두어가 필요없을 정도로 작아야한다.
    - 멤버 변수를 다른 색상으로 표시하거나 눈에 띄게 보여주는 IDE를 사용하자.
  - 인터페이스 클래스와 구현 클래스
    - 인터페이스 이름은 접두어를 붙이지 않는 편이 좋다.
    - 인터페이스 클래스 이름과 구현 클래스 이름 중 하나를 인코딩해야 한다면 구현 클래스 이름을 택하자.
- 자신의 기억력을 자랑하지 마라
  - 명료함이 최고이다. 남들이 이해하는 코드를 내놓는 것이 전문 프로그래머이다,.
- 클래스 이름
  - 클래스 이름, 객체 이름은 명사나 명사구가 적합하다.

- Manager, Processor, Data, Info 등의 단어는 피한다.
- 동사는 사용하지 않는다.
- 메서드 이름
  - 메서드 이름은 동사나 동사구가 적합하다.
  - 접근자, 변경자, 조건자는 javabeen 표준에 따라 값 앞에 get, set, is를 붙인다.
  - 생성자를 overload할 때는 정적 팩토리 메서드를 사용한다.

```
Complex fulcrumPoint = Complex.FromRealNumber(23.0); //bad
Complex fulcrumPoint = new Complex(23.0); //good
```

- 생성자 사용을 제한하려면 해당 생성자를 private로 선언한다.
- 기발한 이름은 피하라
  - 재미난 이름보다 명료한 이름을 선택하라.
  - 특정 문화에서 사용하는 농담을 피하라.
  - 의도를 분명하고 솔직하게 표현하라.
- 한 개념에 한 단어를 사용하라
  - 추상적인 개념 하나에 단어 하나를 선택해 이를 고수하라.
    - 똑같은 메서드를 클래스마다 fetch, retrieve, get으로 제각각 부르면 혼란스럽다.
  - 메서드 이름은 독자적이고 일관적이어야한다.
- 말장난을 하지 마라
  - 한 단어를 두 가지 목적으로 사용하지 마라.
    - 기존 구현한 add 메서드는 기존 값 두 개를 더하거나 이어서 새로운 값을 만든다.
    - 새로 작성하는 메서드는 집합에 값 하나를 추가하는 것이라면, 기존 메서드와 맥락이 다르므로 insert나 append라는 이름이 적당하다.
- 해법 영역에서 가져온 이름을 사용하라
  - 코드를 읽는 사람도 프로그래머다. 전산 용어, 알고리즘 이름, 패턴 이름, 수학 용어 등을 사용해도 괜찮다.
- 문제 영역에서 가져온 이름을 사용하라
  - 적절한 '프로그래머 용어'가 없다면 문제 영역에서 이름을 가져온다.
    - 코드를 보수하는 프로그래머가 분야 전문가에게 의미를 물어 파악할 수 있다.
  - 우수한 프로그래머와 설계자라면 해법 영역과 문제 영역을 구분할 줄 알아야 한다.
    - 문제 영역 개념과 관련이 깊은 코드라면 문제 영역에서 이름을 가져와야한다.
- 의미 있는 맥락을 추가하라
  - 클래스, 함수, 이름 공간에 넣어 맥락을 부여한다.
- 불필요한 맥락을 추가하라
  - 일반적으로는 짧은 이름이 긴 이름보다 좋다. 단, 의미가 분명한 경우에 한해서다. 이름에 불필요한 맥락을 추가하지 않도록 한다.
- 불필요한 맥락을 없애라
  - accountAddress, customerAddress는 Address 클래스 인스턴스로는 좋은 이름이나 클래스 이름으로는 적합하지 못하다.
    - Address는 클래스 이름으로 적합하다.