2장 의미있는 이름

* 들어가면서
  + 소프트웨어의 이름은 어디나 쓰인다.
  + 변수, 함수, 인수와 클래스, 패키지에도 이름을 붙인다.
  + 소스파일에도 이름을 붙이고 소스파일에 담긴 디렉터리도 이름을 붙인다.
  + 이렇듯 여기저기에 직접 이름을 붙여야 하는 상황이 나타나는데 이름을 짓는 간단한 규칙을 소개한다.
* 의도를 분명하게 밝혀라
  + 아무 변수이름은 그 변수가 무슨 역할을 하는지 모름
  + 따라서 의도가 드러나는 명칭을 사용하는 것이 좋다
  + 코드가 하는 일을 짐작하게 해야한다.
  + 코드의 맥락이 함축되지 않고 맥락이 그대로 들어나게 끔 해야한다.
* 그릇된 정보를 피한다.
  + 코드에 그릇된 단서를 남겨서는 안된다. 그릇된 단서는 코드의 의미를 흐린다.
  + 서로 흡사한 이름을 사용하지 않도록 한다.
  + 유사한 개념은 유사한 표기법을 사용한다. 일관성이 떨어지면 그릇된 정보이다.
* 의미 있게 구분해라
  + 컴파일러나 인터프리터만 통과하려는 생각으로 구현하는 프로그램들은 스스로 문제를 일으킨다.
  + 연속된 숫자나 불용어를 추가하는 방식은 적절하지 못하다., 이름이 달라야 한다면 의미도 달라져야 한다.
* 발음하기 쉬운 이름을 사용하라
  + 단어에 능숙하기 떄문에 상당부분은 개념만 전적으로 처리한다. 그리고 정의상으로 단어는 발음이 가능
  + 근데 발음하기 어려운 이름은 토론하기도 어렵다는 단점이 있다.
* 검색하기 쉬운 이름을 사용하라
  + 문자 하나를 사용하는 이름과 상수는 텍스트 코드에 쉽게 눈이 띄지 않는다는 문제점이 있다.
  + 이름의 길이는 범위 크기에 비례해야 한다.
  + 예를들어 sum이라는 단어는 별로 유용하지 않지만, 최소한 검색이 가능하다
* 인코딩을 피해라
  + 이름에 인코딩할 정보는 많다. 유형이나 범위 정보까지 인코딩에 넣으면 그만큼 이름을 해독하기가 어렵다.
  + 추가로 인코딩 언어까지 익히라는 요구는 비합리적
  + 헝가리식 표기법을 피하라, 멤버변수에 접두형을 넣는 것을 피하라, 인터페이스 클래스 이름과 구현클래스 이름에 인코딩을 해야 한다면 구현클래스 이름에 인코딩을 하도록 하자.
* 자신의 기억력을 자랑하지 마라
  + 변수명을 작성하는데있어서 자기가 아는 이름으로 변환했다면 이는 잘못된 것이다.
* 클래스 이름
  + 클래스 이름과 객체 이름은 명사나 명사구가 적합
  + 기존에가지고 있었던 클래스명을 피하고, 동사는 사용하지 않는다
* 메서드 이름
  + 메서드 이름은 동사나 동사구가 적합하다.
  + 접근자, 변경자, 조건자는 javabean 표준에 따라 값에 get set is를 붙인다.
  + 생성자를 중복 정의할 때는 정적 팩토리 메서드를 사용한다.
  + 생성자 사용을 제한하려면 해당 생성자를 private로 선언한다.
* 기발한 이름은 피하라
* 한 개념아 한 단어만 사용한다.
  + 똑 같은 메서드르 클래스마다 fetch, receive, get으로 제각각 부르면 혼란스럽다.
  + 따라서 한 개념에 한 단어만 사용하도록 하는 것이 좋다
* 말장난을 하지 마라
* 해법 영역에서 가져온 이름을 사용하라
* 문제 영역에서 가져온 이름을 사용하라
* 의미있는 맥락을 추가하라
* 불필요한 맥락을 제가하라