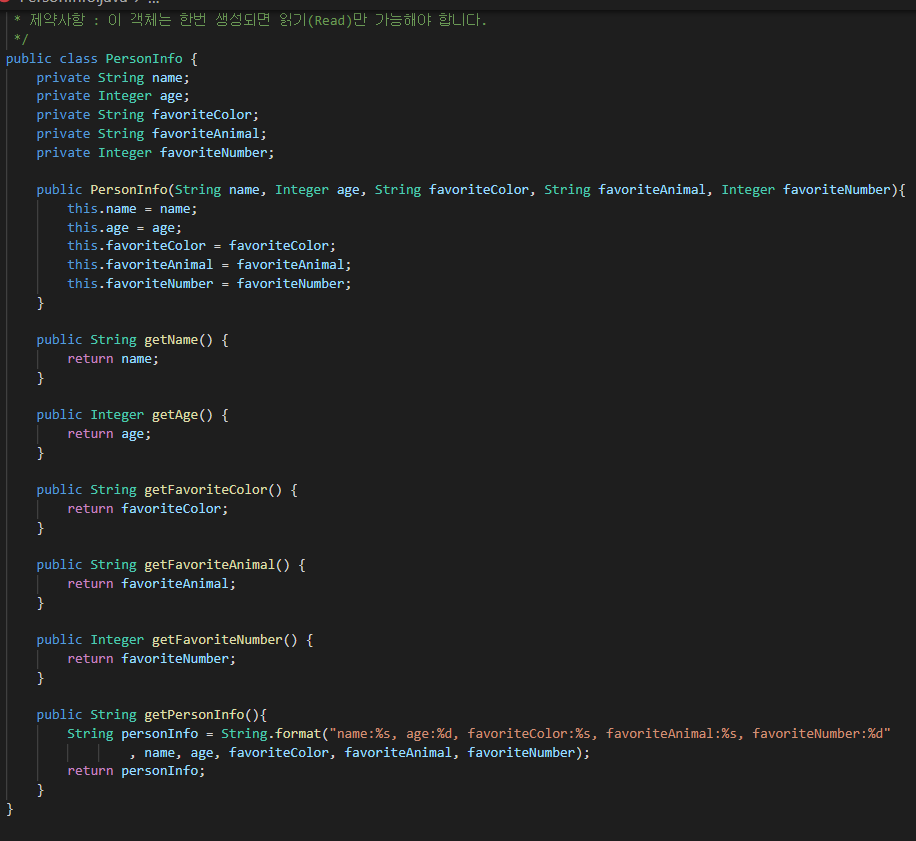
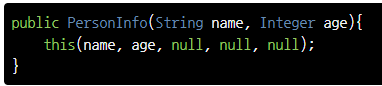
빌더 패턴

참고자료 : <https://jdm.kr/blog/217>

* 빌더패턴이란?
  + 추상 팩토리 패턴이나 팩토리 메소드 패턴과는 조금 다르다. 빌더 패턴도 새로운 객체를 만들어서 반환하는 패턴이긴 하나, 실제 동작하는 방식은 조금 다르다.
  + 빌더 패턴은 생성자에 들어갈 매개 변수가 많든 적든 차례차례 매개 변수를 받아들이고 모든 매개 변수를 받은 뒤에 이 변수들을 통한해서 한번에 사용을 한다.
* 왜 빌더패턴을 사용하나
* 
* 이 클래스는 한 사람의 정보를 담기 위한 것이다. 하지만 제약 사항이 걸려있다
* 따라서 setter메소드는 존재할 수 없고 오로지 생성자를 통해서만 데이터를 입력받는다는 가정으로 만든 클래스이다. 생성자를 보면 모든 매개변수를 받도록 하고 있다. 하지만 어떤 사람들은 모든 데이터가 없을 수도 있다.
* 예를 들어 이름과 나이만 알고 있는 사람의 경우는 다음처럼 객체를 만들어 내야 할 것이다.



* 하지만 이 경우는 기독성이 안좋다, 그래서 추가 생성자를 만들어서 처리하도록 하면 다음과 같은 생성자가 추가될 것이다.



* 이렇게 해서 하나의 케이스에 대해서는 처리는 했다. 하지만 여기에 새로운 유형의 정보가 들어온다면?
* 예를 들면 이름만 있다거나, 이름과 좋아하는 동물만 있다거나 한다면 그때마다 새로운 생성자를 만들어 줘야 한다.
* 또한 생성자 내부에서 데이터를 입력하는 순서 또한 정해져 있기 때문에 순서 잠깐 바꾼다면 큰일날 수도 있다.

빌더패턴을 적용하기

* 해결해야 할 것은 아래와 같다
  + 불필요한 생성자를 만들지 않고 객체를 만든다
  + 데이터의 순서에 상관없이 객체를 만들어 내야 한다
  + 사용자가 봤을 때 명시적이고 이해할 수 있어야 한다
* 그래서 다음과 같은 빌더 패턴을 적용한 빌더 클래스가 필요한 것이다.

