

16장. SerialDate 리팩토링

😊 SerialDate 클래스의 코드를 리팩토링

SerialDate 특징

1. `java.util.Date`와는 다르게 시간은 나오지 않는다. 예) 2023-08-01
 - a. `java.util.Date`는 시간까지 나옴
2. serializable (직렬화)
 - a. 객체를 바이트 단위로 변환

David Gilbert가 2001년에 만든 클래스

Uncle Bob은 왜 SerialDate를 리팩토링하는가

1. 클린코드는 2008년에 초판 인쇄
2. 실행 가능한 문장 185개중 단위테스트가 실행하는 문장이 91개 불과
3. 주석 처리 된 많은 테스트 코드
4. 리팩토링하면서 테스트 커버지리도 올리는 예시를 보여줄수 있음

첫째. 돌려보자

주석 처리 되어 있는 테스트 코드

```
// todo assertEquals(-1, stringToMonthCode("0"));  
// assertEquals(-1, stringToMonthCode("13"));
```

stringToMonthCode

```
public static int stringToMonthCode(java.lang.String s)
Converts a string to a month code.

Parameters:
s - the string to parse. (예: "1", "2", ..., "12")
Returns:
-1 if the string is not parseable, the month of the year otherwise.
```

stringToMonthCode 구현 부분

```
public static int stringToMonthCode(String s) {
439
    // jan, feb, .... dec
440    final String[] shortMonthNames = DATE_FORMAT_SYMBOLS.getShortMonths();
    // january, february, .... december
441    final String[] monthNames = DATE_FORMAT_SYMBOLS.getMonths();
442
443    int result = -1;
444    s = s.trim();
445
446    // first try parsing the string as an integer (1-12)...
447    try {
448        result = Integer.parseInt(s);
449    }
450    catch (NumberFormatException e) {
451        // suppress
452    }
453
454    // now search through the month names...
455    if ((result < 1) || (result > 12)) {
        result = -1
456        for (int i = 0; i < monthNames.length; i++) {
457            if (s.equals(shortMonthNames[i])) {
458                result = i + 1;
459                break;
460            }
461            if (s.equals(monthNames[i])) {
462                result = i + 1;
463                break;
464            }
465        }
466    }
467
468    return result;
}
```

```
469  
470 }
```
