Victoria University
of Wellington, New Zealand
Te Whare Wananga o te
Upoko o te Ika a Maui
Aotearoa

SWEN221: Software Development

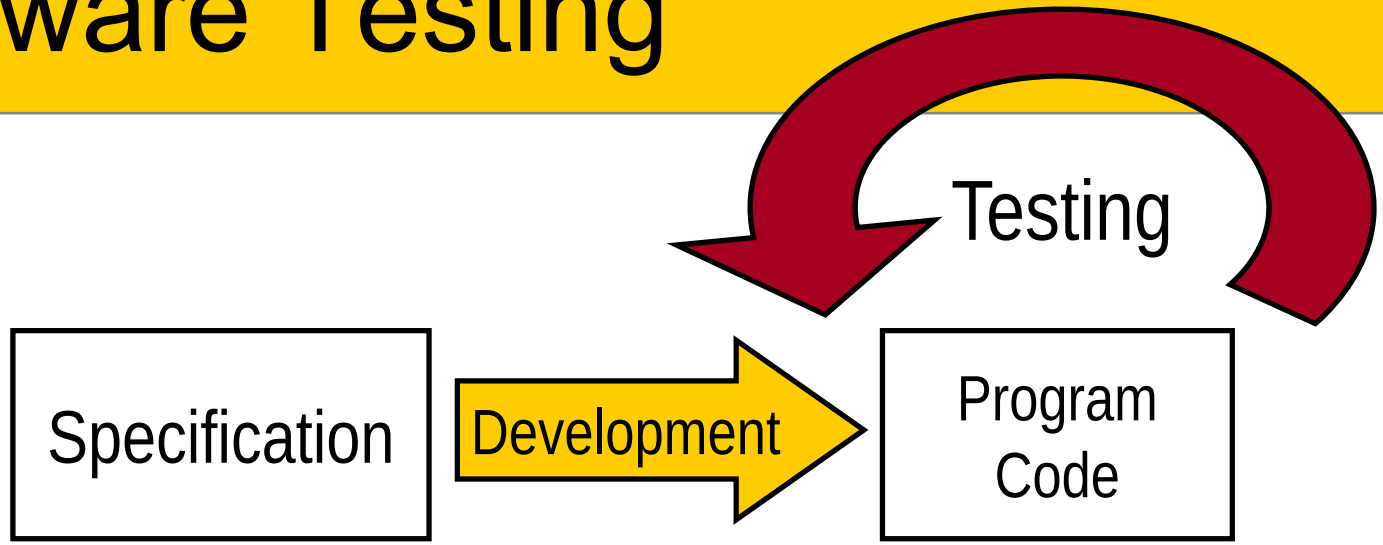
#2 – Testing I

David J. Pearce & Marco Servetto
Computer Science, Victoria University

Testing

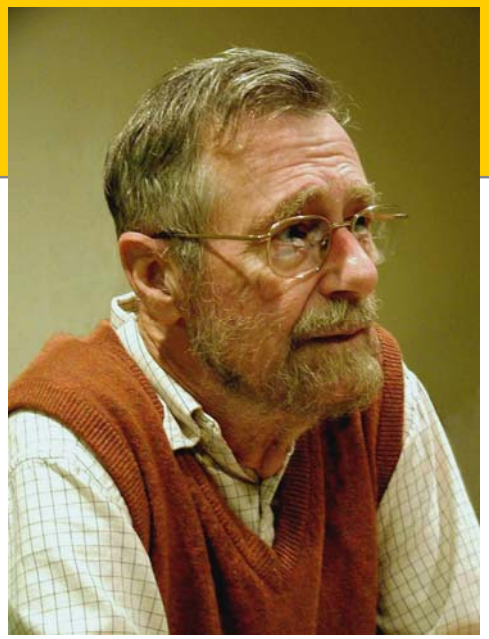
Why Test?

Software Testing



- Why test?
  - Code never works first time!
  - You must test it to find the bugs!
  - But, what is a bug?
    - Obvious ones e.g. divide-by-zero
    - Subtle failures to meet specification
  - Testing only increases confidence in software
    - It cannot guarantee there are no bugs

Testing



Edsger Wybe Dijkstra:

"Program testing can be used to show the presence of bugs, but never to show their absence!"

`http://www.cs.utexas.edu/users/EWD/`

What testing cannot do

- Unfortunately, testing cannot be ***exhaustive***

```
boolean isPrime(int x) {
 …
}
```

  - Has 2^{32} possible inputs.
  - If each test takes 1 second then exhaustive test takes:

  - Must pick out *test cases* to represent input domain

Unit testing with **JUnit**

JUnit a Unit Testing Framework

- **Kent Beck** (XP, Smalltalk)
- **Erich Gamma** (Eclipse, Patterns)

Using Junit:

- Tests are Java methods
- Test suites are Java classes
- Annotations mark them out
- API for writing tests
- IDE support (Eclipse…)

- http://junit.sourceforge.net/

Anatomy of a JUnit Test

In your test class

- (typically 1-1 with application classes)

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
```

- Annotate methods with **@Test**

The JUnit API

A range of assertion methods:

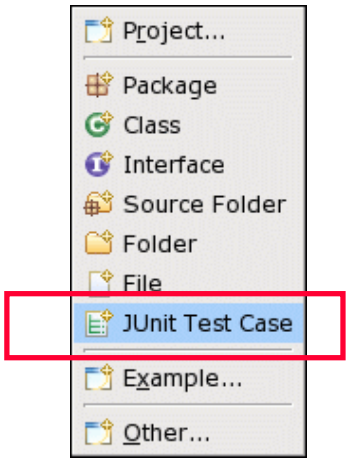
- assertTrue(boolean)
- assertTrue(String message, boolean)

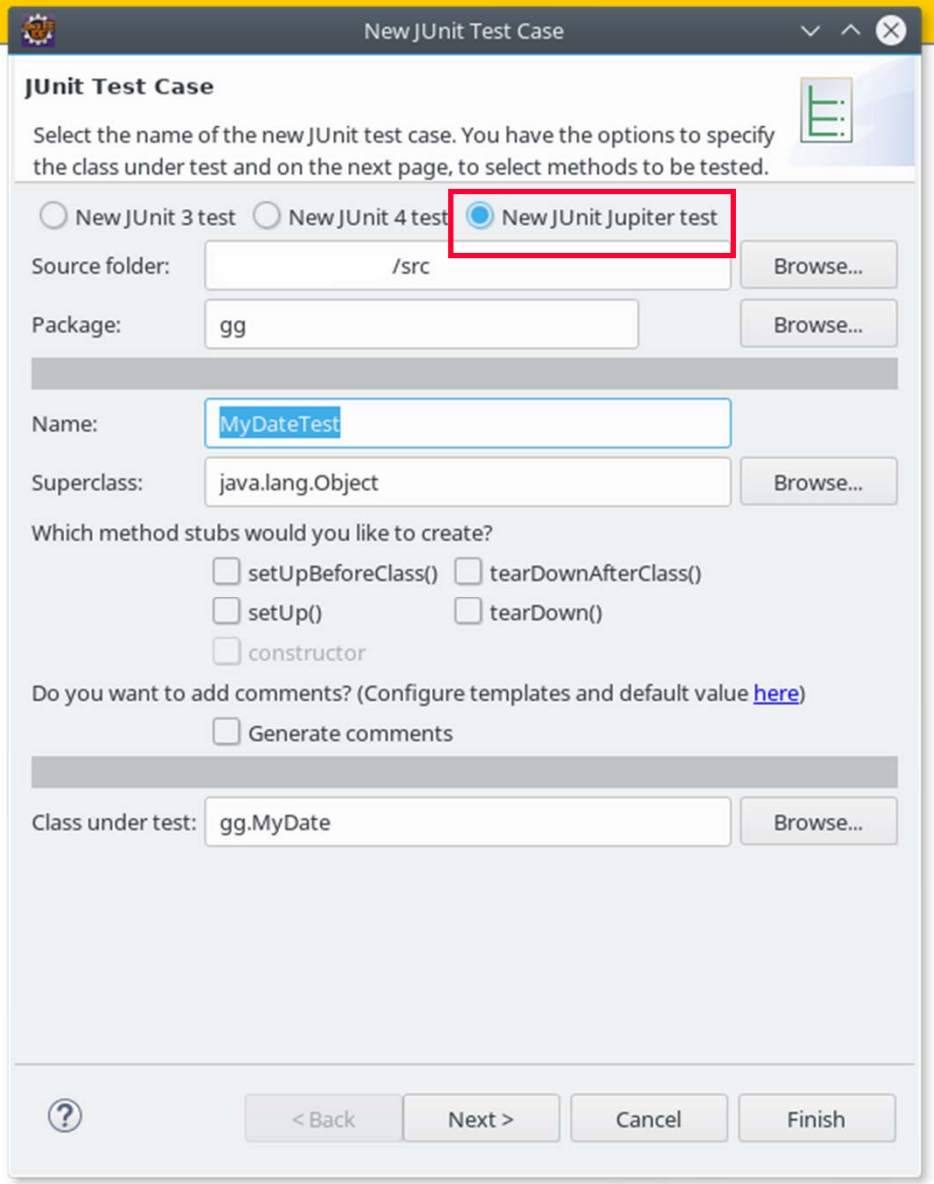
And a whole lot more:

- assertEquals(Object expect, Object actual)
- assertEquals(float expected, float actual, float delta)
- assertNull, assertNotNull
- assertTrue, assertFalse
- assertSame, assertNotSame
- fail(), fail(String message)

```
public class MyDate {
  private int day, month, year; // 1 <= day <= 31 and 1 <= month <= 12

  public MyDate(int day, int month, int year) {
   this.day = day;
   this.month = month;
   this.year = year;

   // check invariantS hold
   if(day <= 0 || month < 0) { throw new RuntimeException(…); }

   else if((month==4 || month==6 || month==9 || month==11) && day > 30) {
    throw new RuntimeException("Cannot construct invalid Date!");

   } else if(month == 2 && (day>29 || (day>28 && !(year%4==0 &&
       (year%100 != 0 || year%400==0))))) {
    throw new RuntimeException("Cannot construct invalid Date!");

   } else if(day > 31 || month > 12) {
    throw new RuntimeException("Cannot construct invalid Date!");
   }
 }

 public int day() { return day; }
 public int month() { return month; }
 public int year() { return year; }
}
```

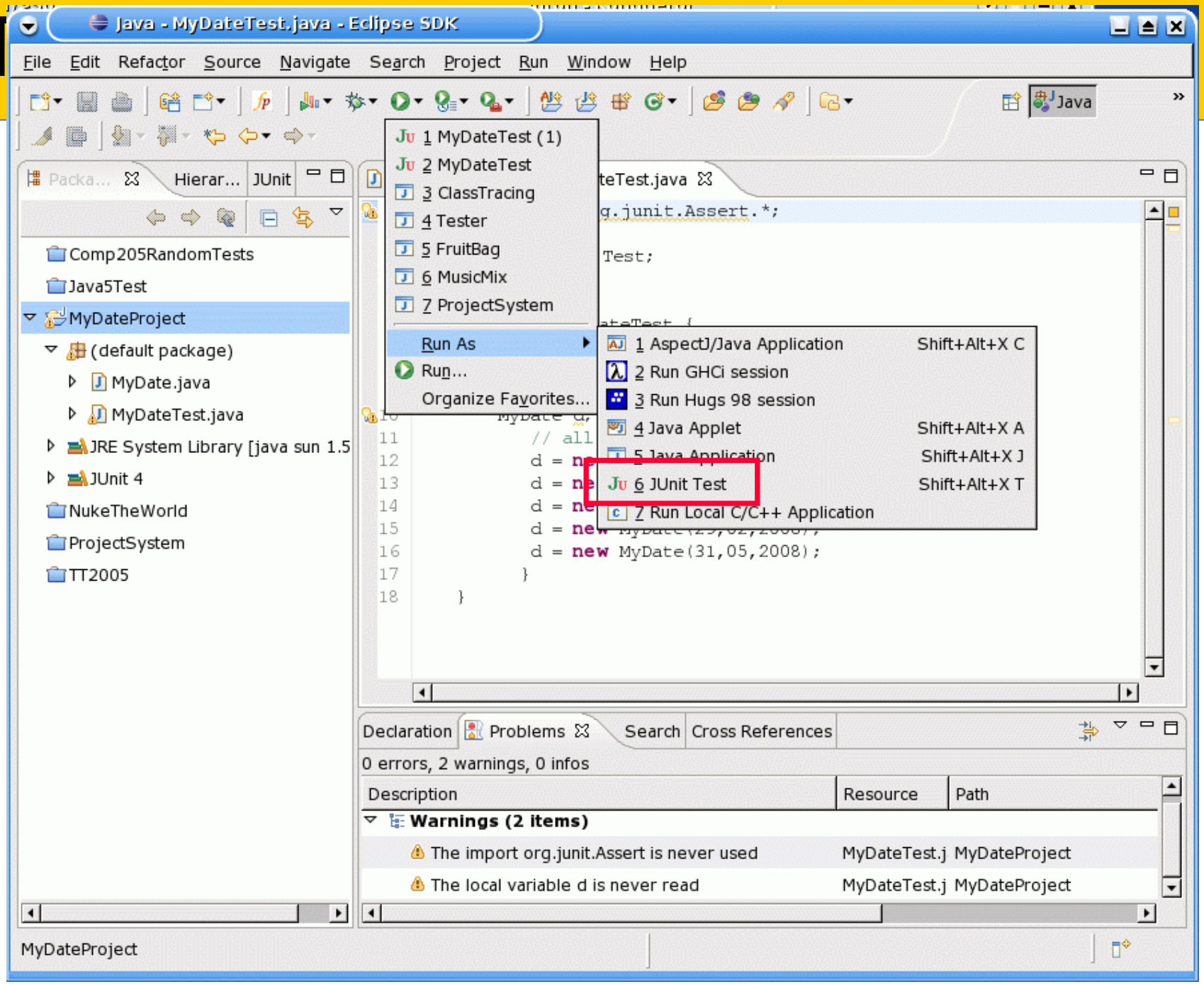
Starting JUnit



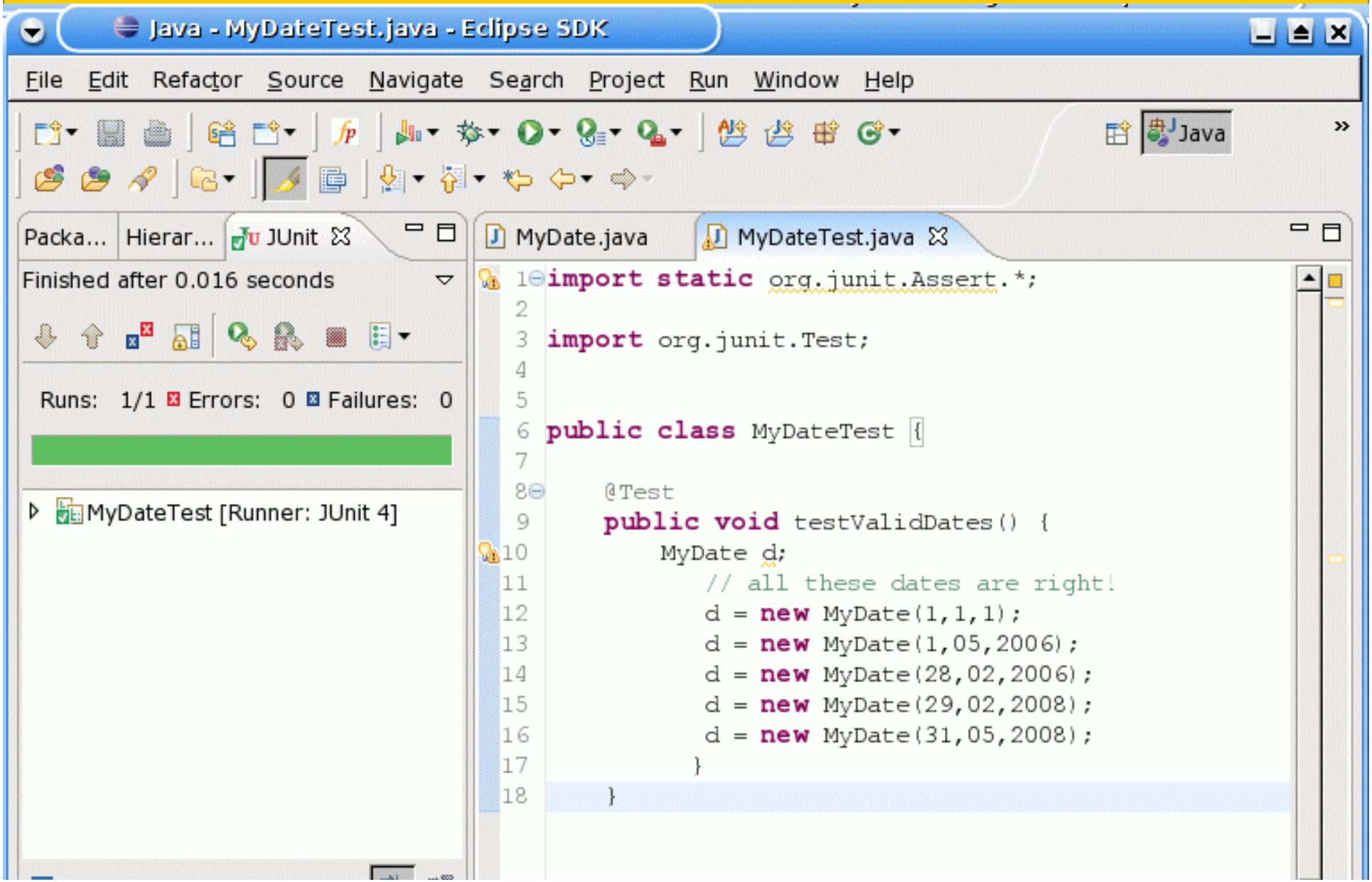


A simple JUnit test

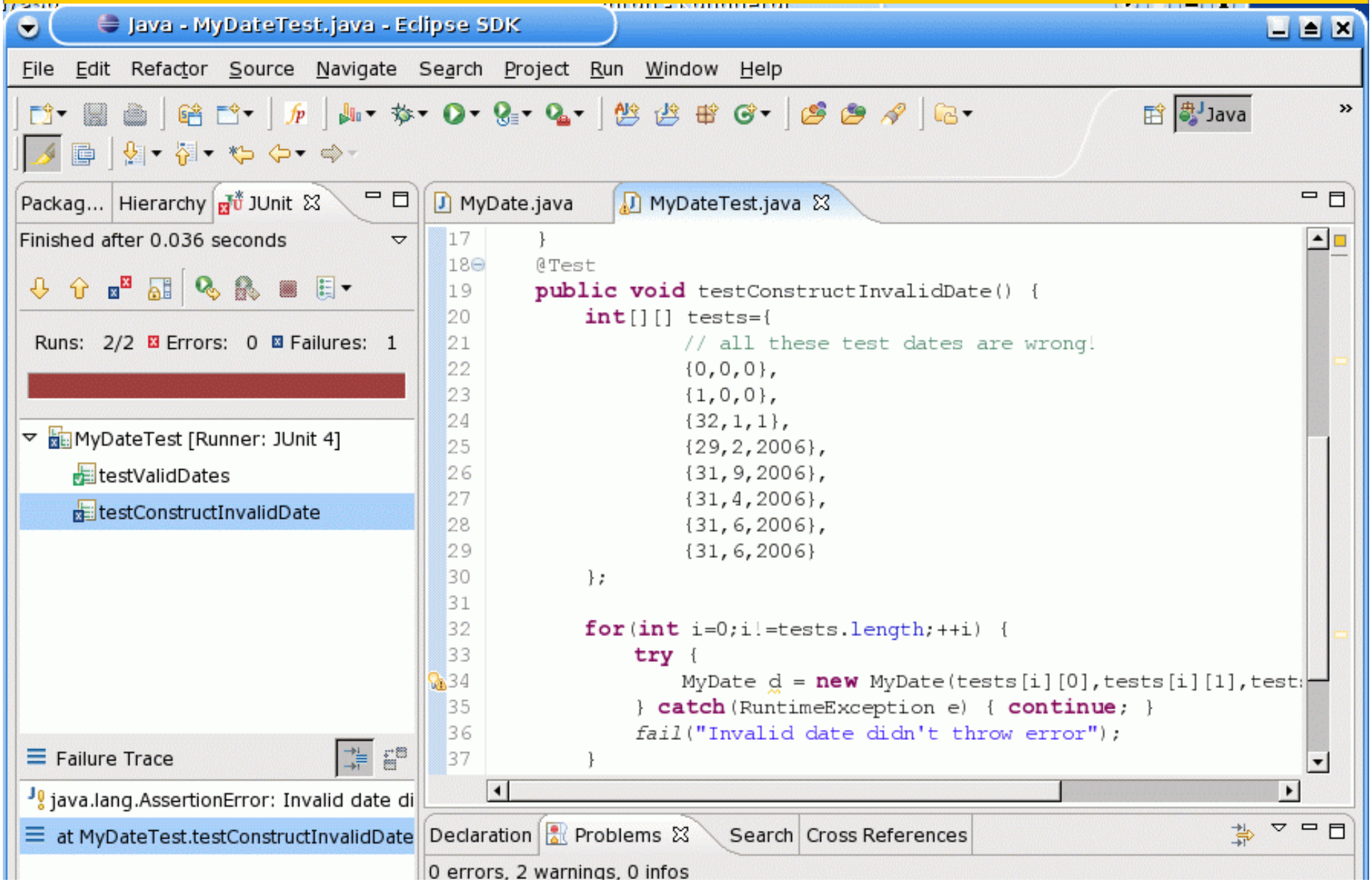
```
public class MyDateTest {
@Test  public void testConstructValidDate() {
   MyDate d;   // all these dates are right!
   d = new MyDate(1,1,1);
   d = new MyDate(1,05,2006);
   d = new MyDate(28,02,2006);
   d = new MyDate(29,02,2008);
   d = new MyDate(31,05,2008);
 }
}
```


Java - MyDateTest.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Run Window Help
Java
1 MyDateTest (1)
2 MyDateTest
3 ClassTracing
4 Tester
5 FruitBag
6 MusicMix
7 ProjectSystem
Run As
Run...
Organize Favorites...
1 AspectJ/Java Application Shift+Alt+X C
2 Run GHCi session
3 Run Hugs 98 session
4 Java Applet Shift+Alt+X A
5 Java Application Shift+Alt+X J
6 JUnit Test Shift+Alt+X T
7 Run Local C/C++ Application
Packa... Hierar... JUnit
Comp205RandomTests
Java5Test
MyDateProject
(default package)
MyDate.java
MyDateTest.java
JRE System Library [java sun 1.5
JUnit 4
NukeTheWorld
ProjectSystem
TT2005
teTest.java
g.junit.Assert.*;
Test;
// all
d = new MyDate(31,05,2008);
Declaration Problems Search Cross References
0 errors, 2 warnings, 0 infos
Description Resource Path
Warnings (2 items)
The import org.junit.Assert is never used MyDateTest.j MyDateProject
The local variable d is never read MyDateTest.j MyDateProject
MyDateProject

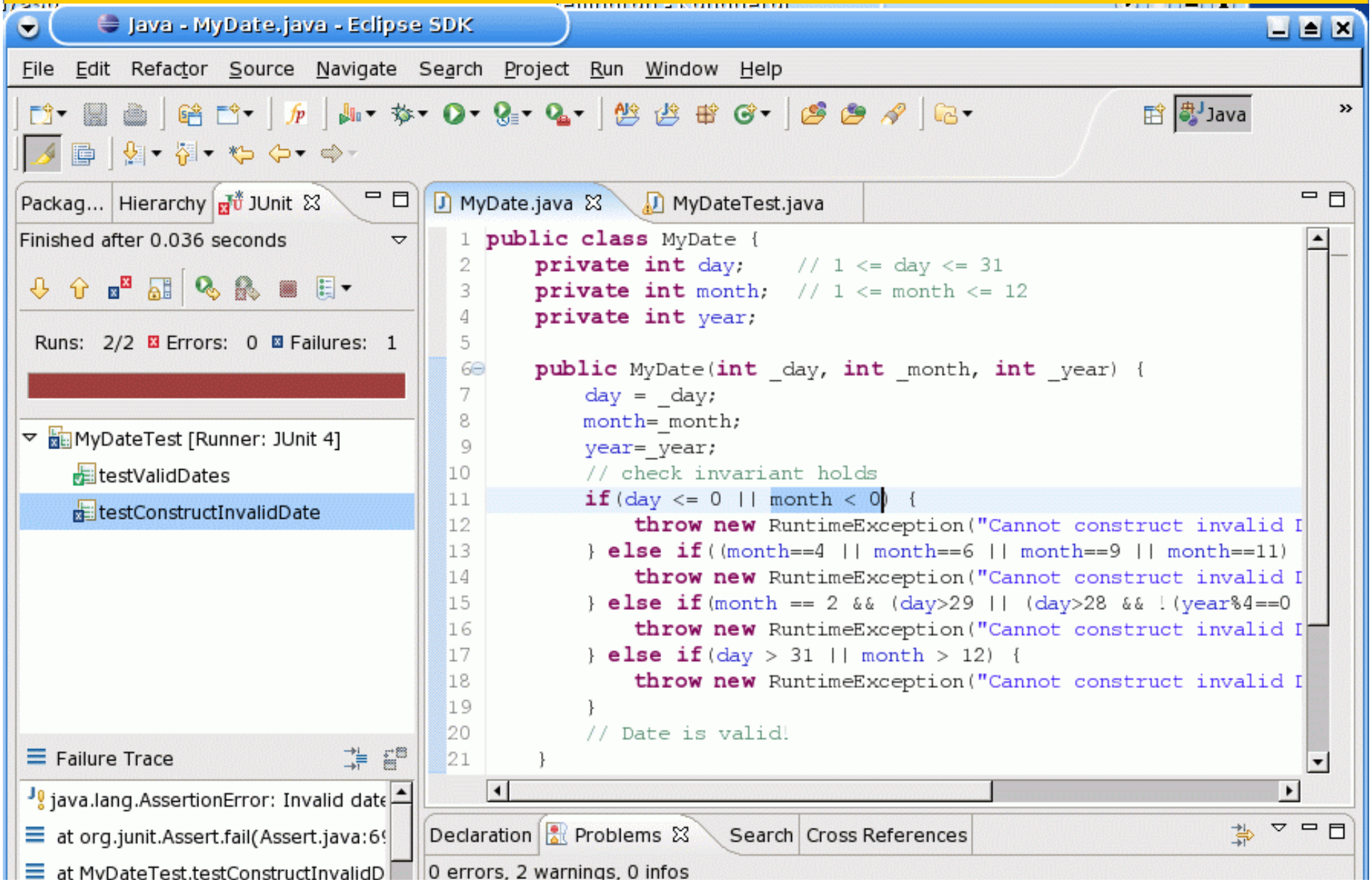
Testing the Happy Path



Testing the Unhappy Path



Why?


Java - MyDate.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Run Window Help
Java
Packag... Hierarchy JUnit
Finished after 0.036 seconds
Runs: 2/2 Errors: 0 Failures: 1
MyDateTest [Runner: JUnit 4]
testValidDates
testConstructInvalidDate
Failure Trace
java.lang.AssertionError: Invalid date
at org.junit.Assert.fail(Assert.java:6
at MyDateTest.testConstructInvalidD
MyDate.java
MyDateTest.java
1 public class MyDate {
2 private int day; // 1 <= day <= 31
3 private int month; // 1 <= month <= 12
4 private int year;
5
6 public MyDate(int _day, int _month, int _year) {
7 day = _day;
8 month=_month;
9 year=_year;
10 // check invariant holds
11 if(day <= 0 || month < 0) {
12 throw new RuntimeException("Cannot construct invalid I
13 } else if((month==4 || month==6 || month==9 || month==11)
14 throw new RuntimeException("Cannot construct invalid I
15 } else if(month == 2 && (day>29 || (day>28 && !(year%4==0
16 throw new RuntimeException("Cannot construct invalid I
17 } else if(day > 31 || month > 12) {
18 throw new RuntimeException("Cannot construct invalid I
19 }
20 // Date is valid!
21 }
Declaration Problems Search Cross References
0 errors, 2 warnings, 0 infos