SWEN221 Software Development

Testing I

Thomas Kuehne

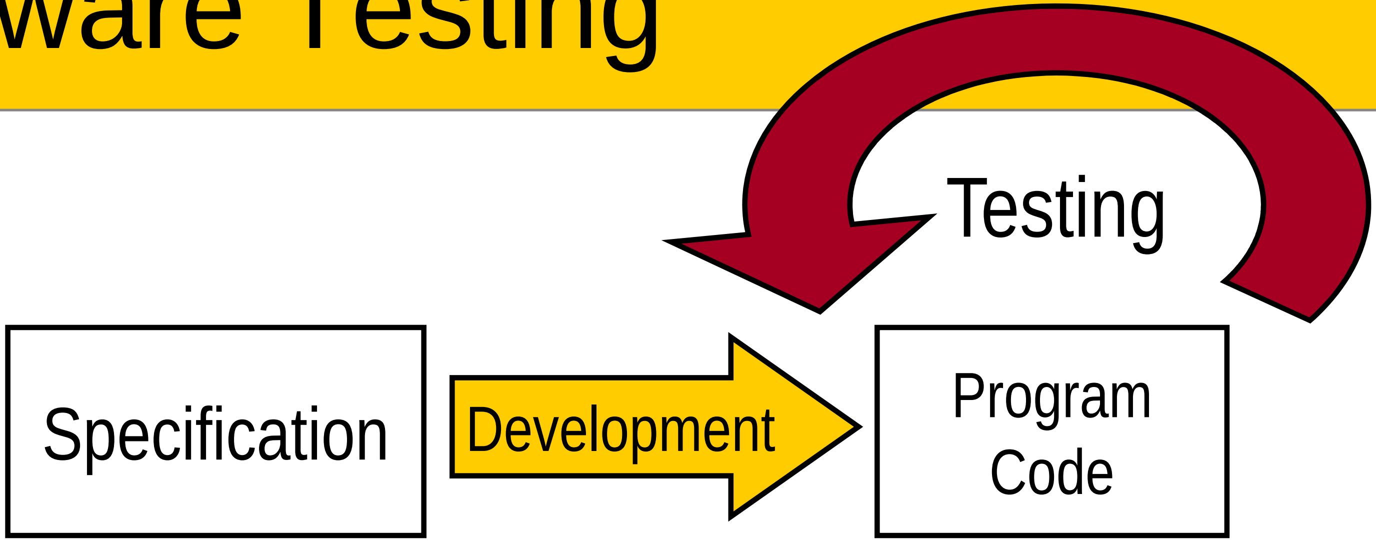
Victoria University

(slides modified from slides by David Pearce)

Testing

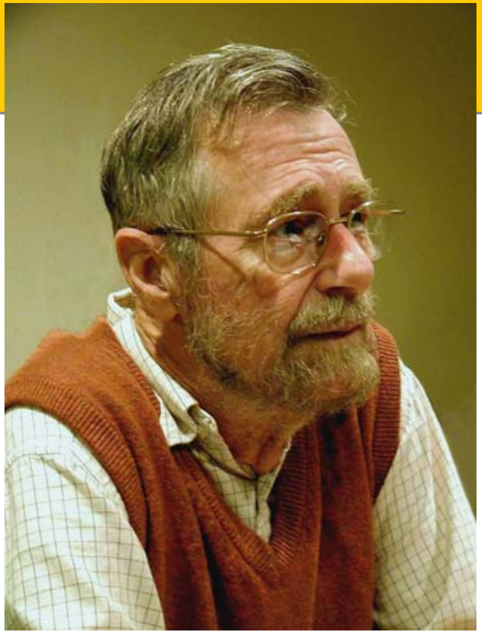
Why Test?

Software Testing



- Why test?

Testing



Edsger W. Dijkstra
(1930 – 2002)

"*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:

Unit testing with **JUnit 4**

JUnit 4 a Unit Testing Framework
- **Kent Beck** (XP, Smalltalk)
- **Erich Gamma** (Eclipse, Patterns)

Using Junit:
- Tests are written in Java methods
- Test suites are Java classes
- Annotations mark them out
- Rules for writing tests
- IDE support (Eclipse…)

- http://junit.sourceforge.net/

Anatomy of a **JUnit 4** Test

In your test classes

– (typically paired 1-1 with application classes)

- import **static org.junit.Assert.*;**
- import **org.junit.***

- Annotate methods with **@Test**

The JUnit 4 Infrastructure

A range of assertion methods, e.g.:

- assertTrue(**Boolean** exp)
- assertTrue(**String** message, **Boolean** exp)

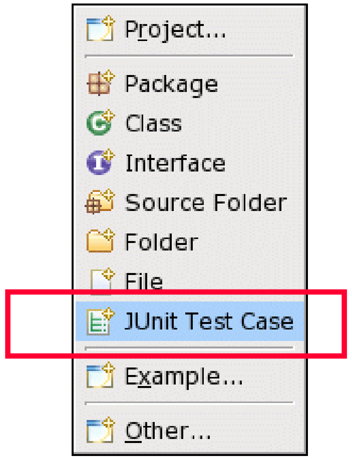
And a whole lot more:

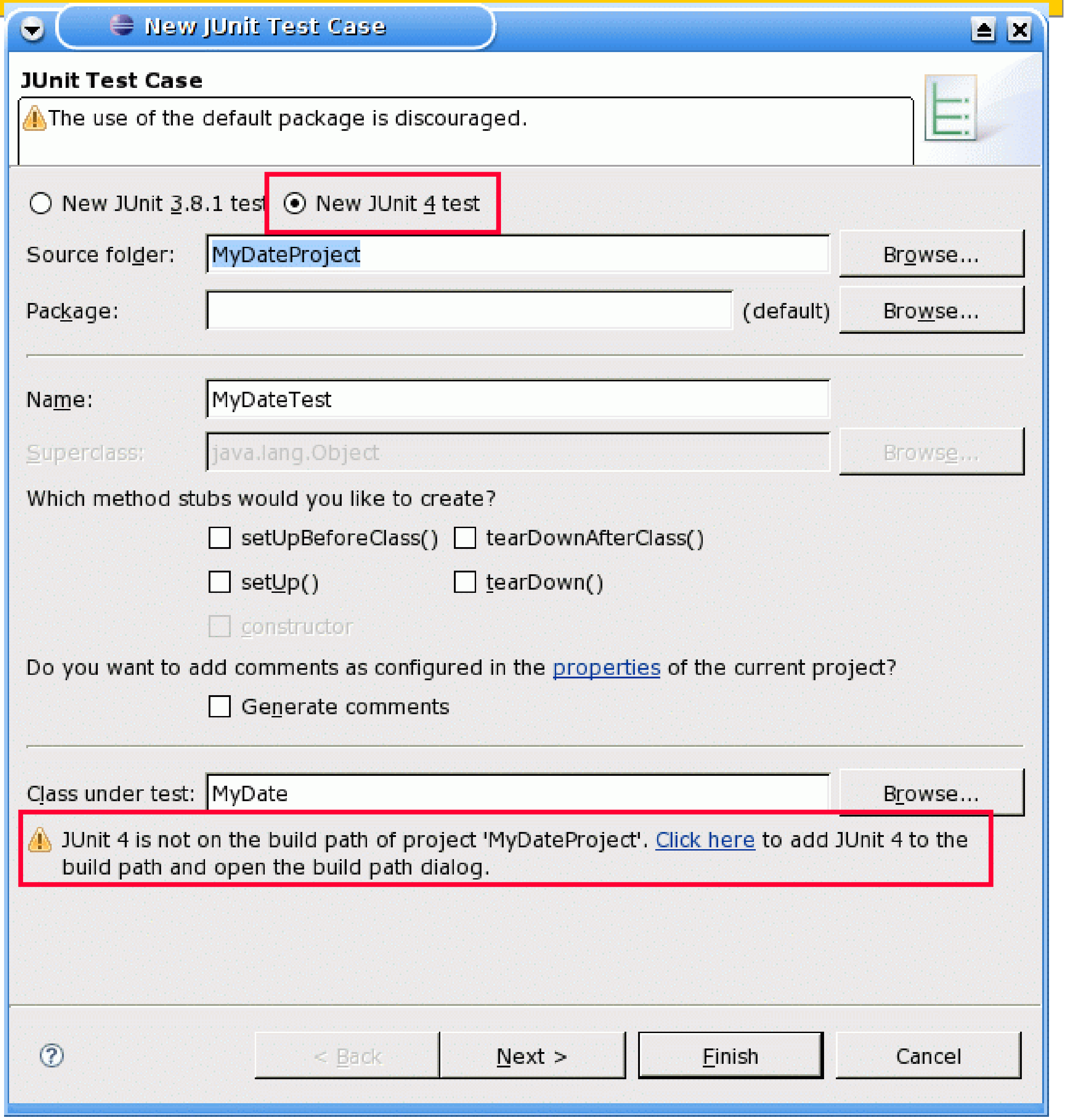
- assertEquals(**Object** expect, **Object** actual)
- assertEquals(**float** expected, **float** actual, **float** delta)
- assertFalse, assertNull, assertNotNull
- assertSame, assertNotSame

- 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 validity of construction parameters
   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 4




New JUnit Test Case

JUnit Test Case

The use of the default package is discouraged.

New JUnit 3.8.1 test [✓ New JUnit 4 test]

Source folder: MyDateProject

Package: (default)

Name: MyDateTest

Superclass: java.lang.Object

Which method stubs would you like to create?

- [ ] setUpBeforeClass()
- [ ] tearDownAfterClass()
- [ ] setUp()
- [ ] tearDown()
- [ ] constructor

Do you want to add comments as configured in the properties of the current project?

- [ ] Generate comments

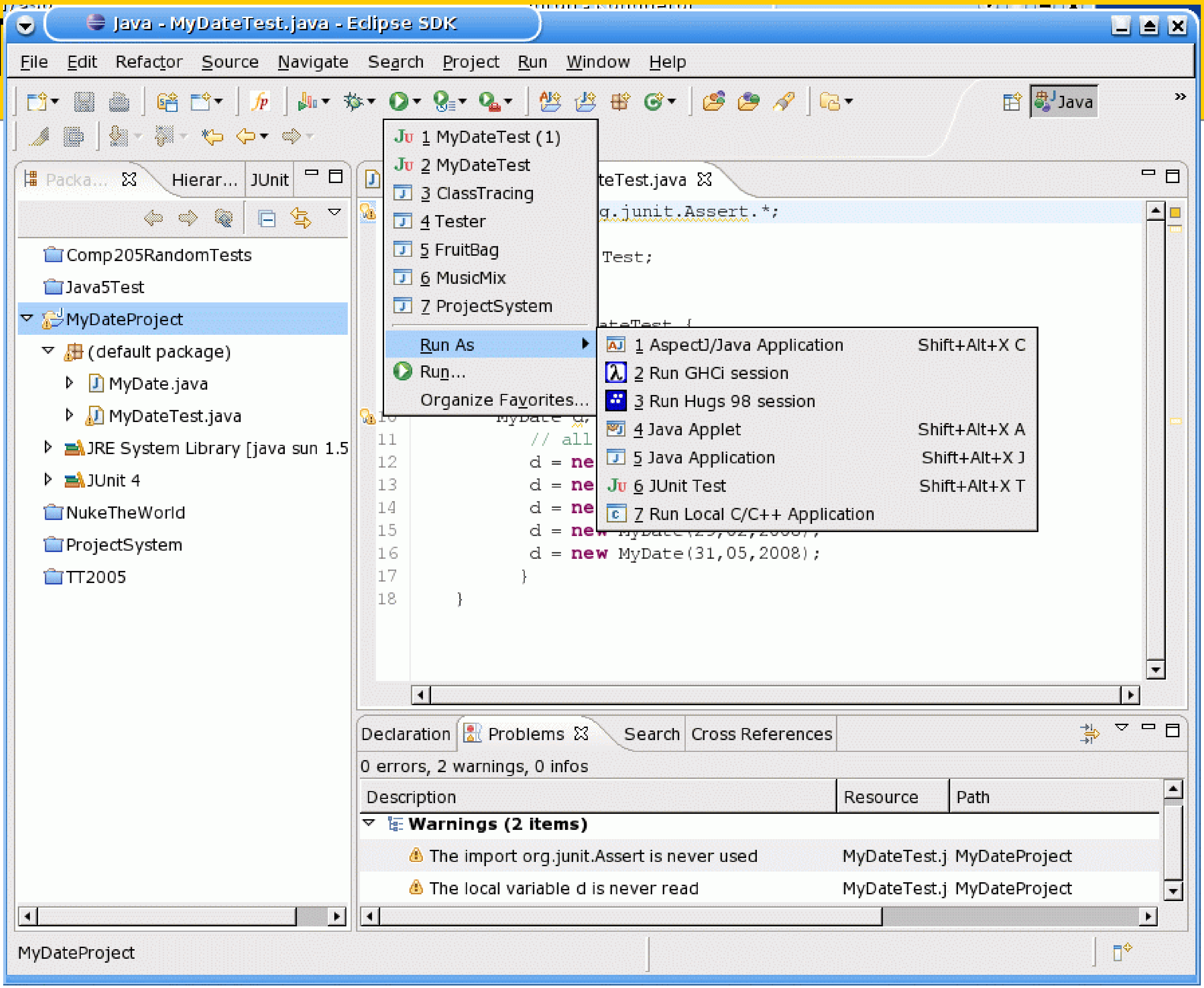
Class under test: MyDate

JUnit 4 is not on the build path of project 'MyDateProject'. Click here to add JUnit 4 to the build path and open the build path dialog.

A simple JUnit test

```
public class MyDateTest {

@Test  public void testValidDates() {
   MyDate d;   // all these dates are valid!

   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);
 }

}
```

T

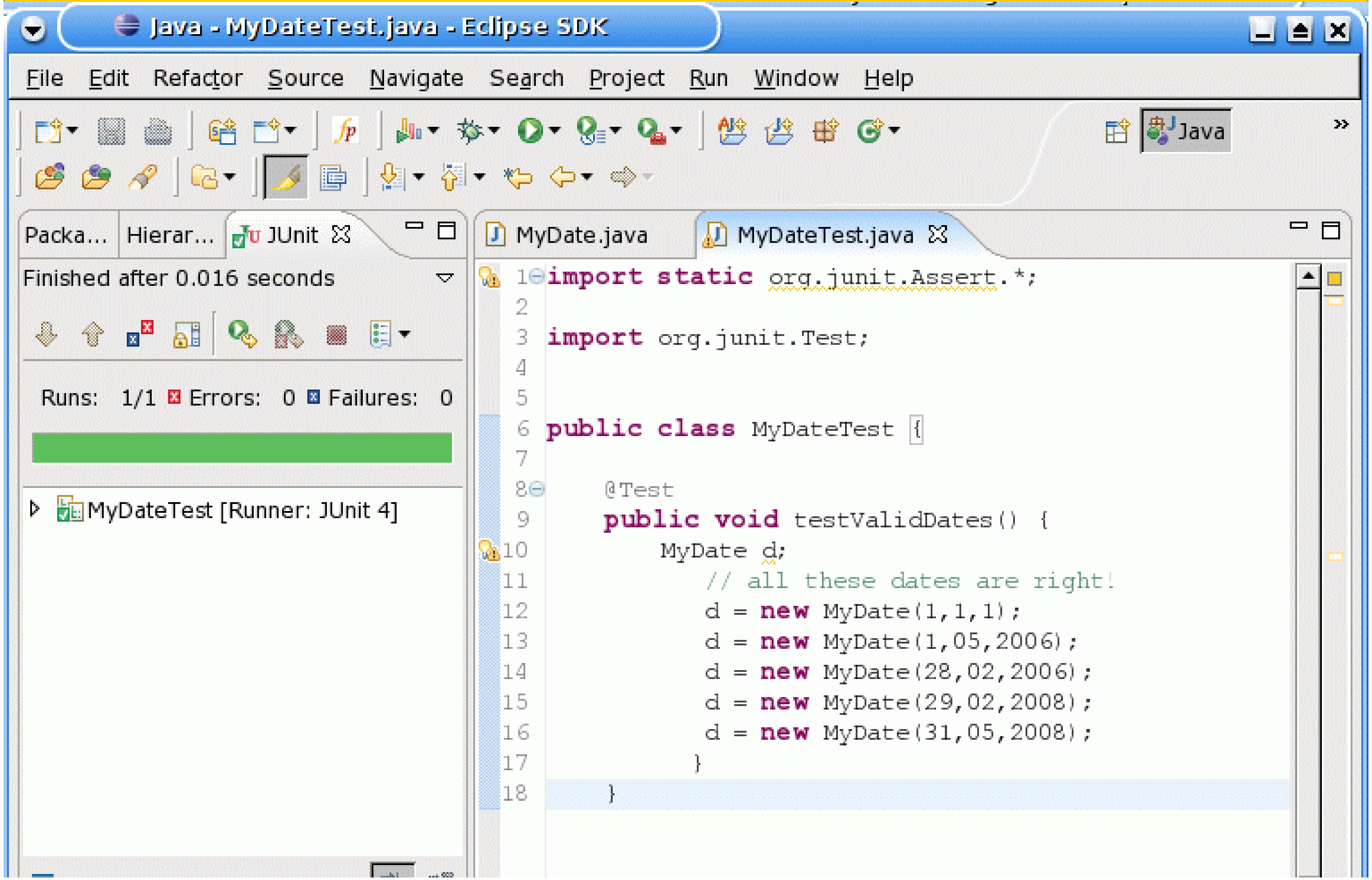


```
g.junit.Assert.*;

Test;

ateTest {

    MyDate d;
        // all
        d = ne
        d = ne
        d = ne
        d = new MyDate(29,02,2008);
        d = new MyDate(31,05,2008);
    }
}
```

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

Testing the Happy Path


Java - MyDateTest.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Run Window Help
Java
Packa... Hierar... JUnit
MyDate.java MyDateTest.java

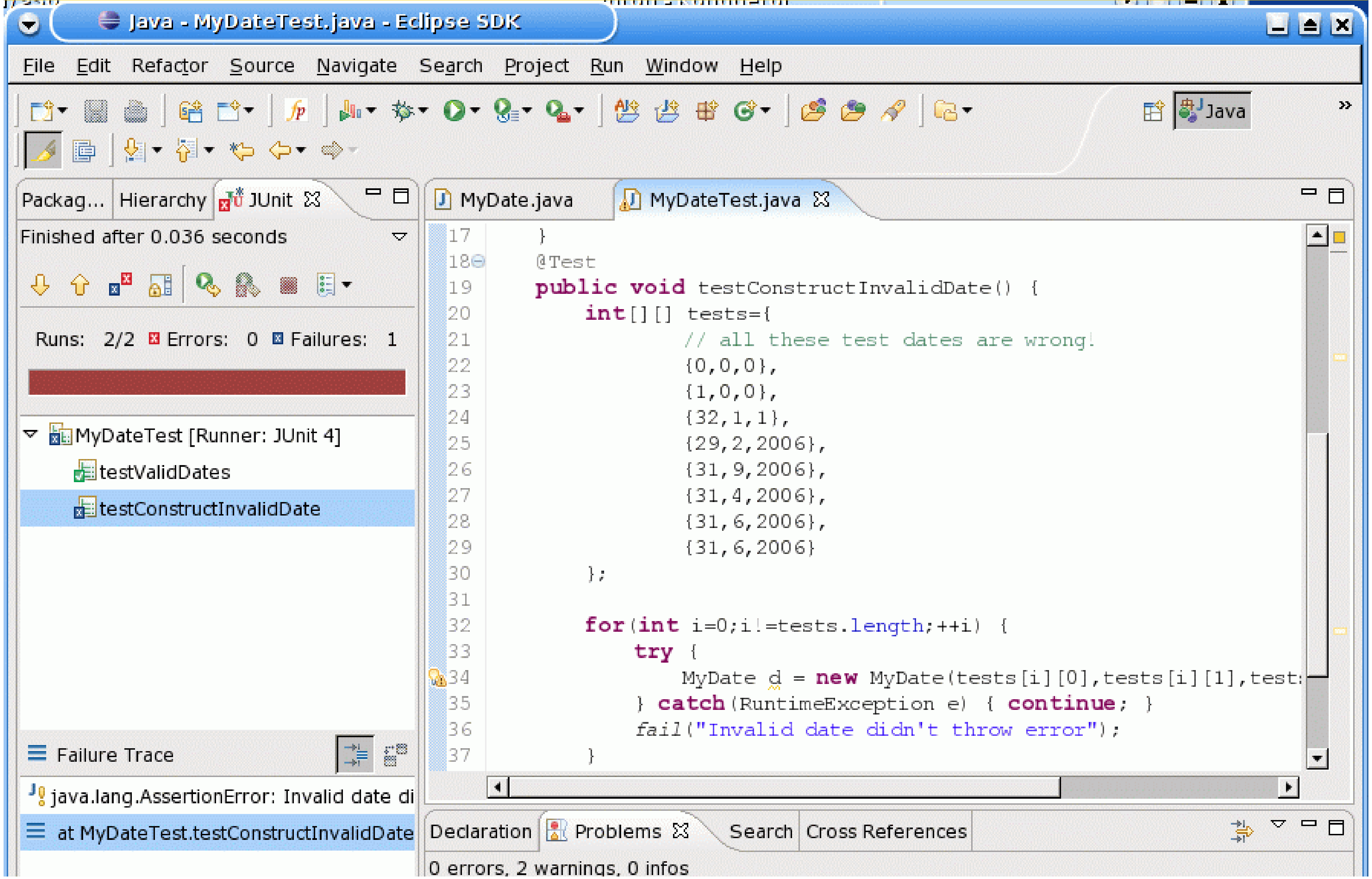
Finished after 0.016 seconds

Runs: 1/1 Errors: 0 Failures: 0

MyDateTest [Runner: JUnit 4]

```java
import static org.junit.Assert.*;

import org.junit.Test;


public class MyDateTest {

    @Test
    public void testValidDates() {
        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);
        }
    }
```

Testing the Unhappy Path


Java - MyDateTest.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Run Window Help
Java
Packag... Hierarchy JUnit
MyDate.java MyDateTest.java
Declaration Problems Search Cross References

Finished after 0.036 seconds

Runs: 2/2 Errors: 0 Failures: 1

MyDateTest [Runner: JUnit 4]
testValidDates
testConstructInvalidDate

Failure Trace

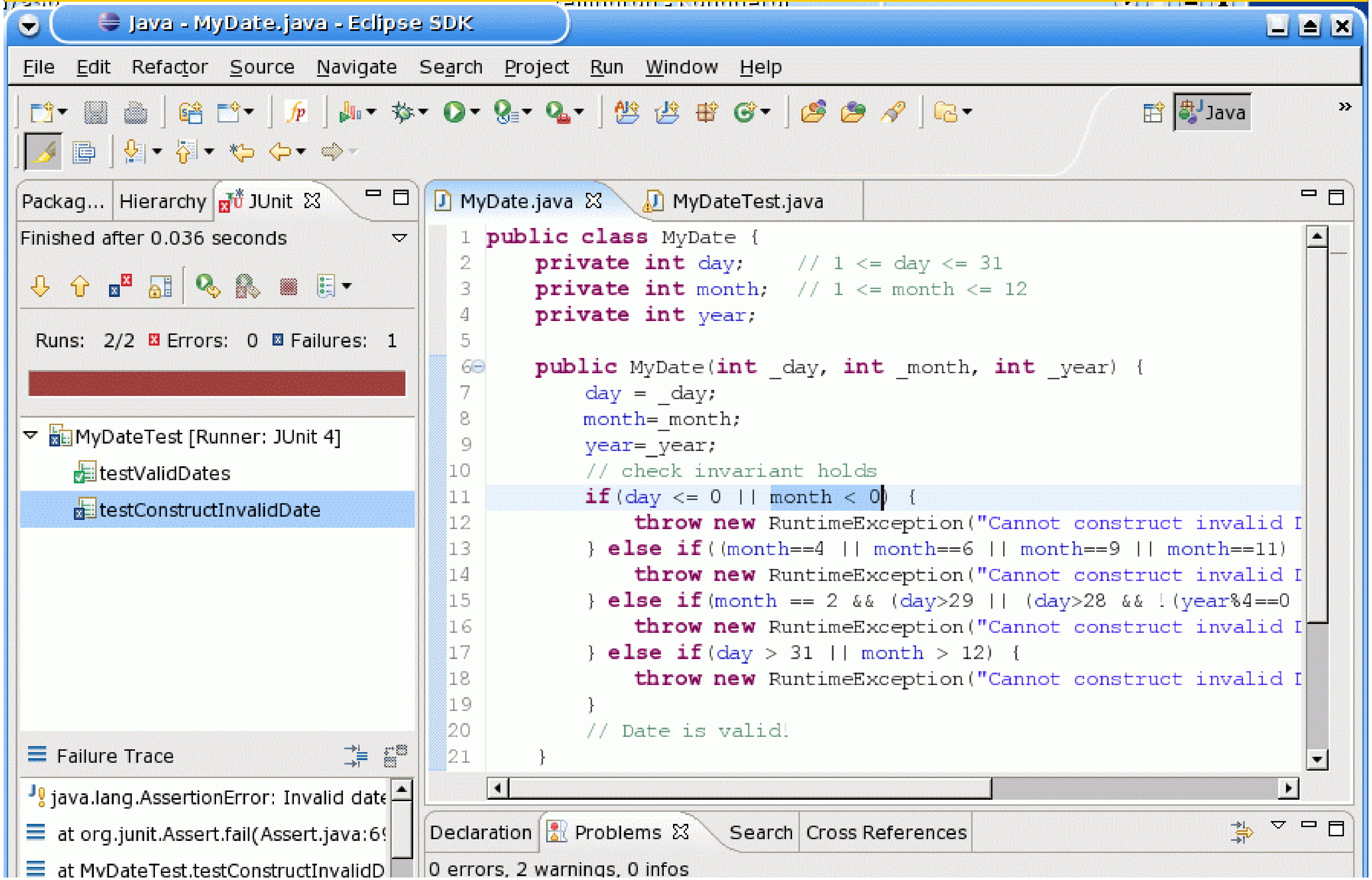
java.lang.AssertionError: Invalid date di

at MyDateTest.testConstructInvalidDate

```
    }
    @Test
    public void testConstructInvalidDate() {
        int[][] tests={
                // all these test dates are wrong!
                {0,0,0},
                {1,0,0},
                {32,1,1},
                {29,2,2006},
                {31,9,2006},
                {31,4,2006},
                {31,6,2006},
                {31,6,2006}
        };

        for(int i=0;i!=tests.length;++i) {
            try {
                MyDate d = new MyDate(tests[i][0],tests[i][1],test
            } catch(RuntimeException e) { continue; }
            fail("Invalid date didn't throw error");
        }
```

0 errors, 2 warnings, 0 infos

Why?



Java - MyDate.java - Eclipse SDK
File Edit Refactor Source Navigate Search Project Run Window Help
Java
Packag... Hierarchy JUnit
MyDate.java MyDateTest.java
Declaration Problems Search Cross References

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
```

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

    public MyDate(int _day, int _month, int _year) {
        day = _day;
        month=_month;
        year=_year;
        // check invariant holds
        if(day <= 0 || month < 0)  {
            throw new RuntimeException("Cannot construct invalid I
        } else if((month==4 || month==6 || month==9 || month==11)
            throw new RuntimeException("Cannot construct invalid I
        } else if(month == 2 && (day>29 || (day>28 && !(year%4==0
            throw new RuntimeException("Cannot construct invalid I
        } else if(day > 31 || month > 12)  {
            throw new RuntimeException("Cannot construct invalid I
        }
        // Date is valid!
    }
```

0 errors, 2 warnings, 0 infos