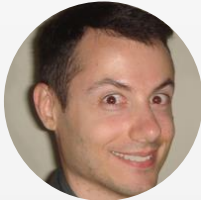


# How Java Programmers Test Exceptional Behavior



Diego Marcilio



Carlo A. Furia



Università  
della  
Svizzera  
italiana

Software  
Institute

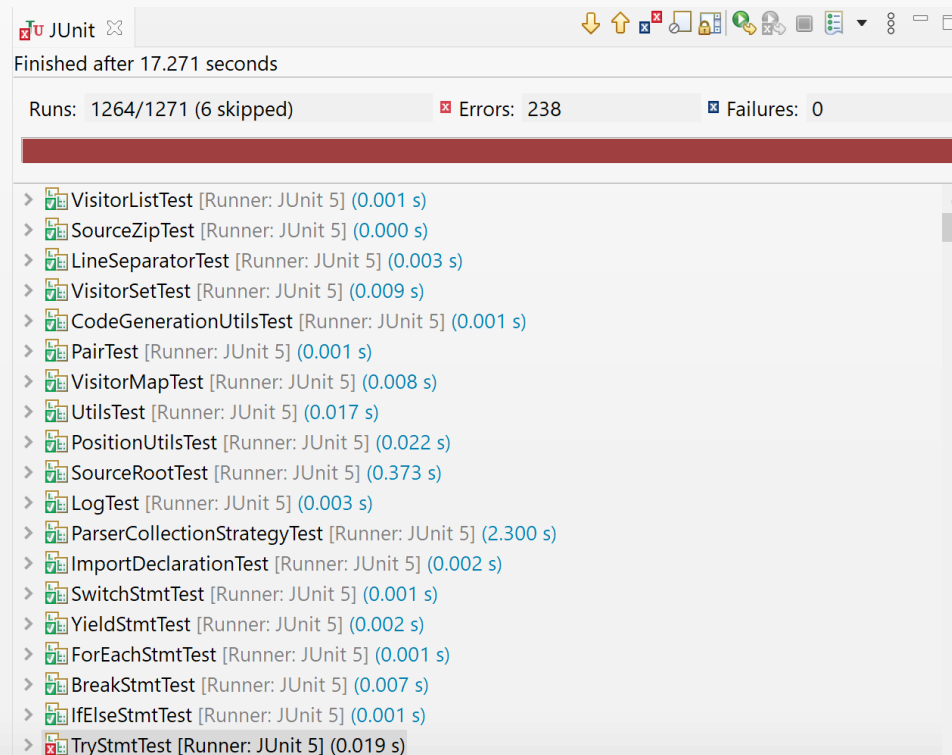


MSR 2021

# Exceptions often signal undesired behavior

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "Not a Number"  
    at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)  
    at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)  
    at java.base/java.lang.Double.parseDouble(Double.java:556)  
    at java.base/java.lang.Double.valueOf(Double.java:519)  
    at org.apache.commons.lang3.math.NumberUtils.createDouble(NumberUtils.java:911)  
    at usi.atom.NumberFormatter.createDouble(NumberFormatter.java:8)  
    at usi.atom.NumberFormatter.main(NumberFormatter.java:12)
```

# Tests are created to find bugs





 [junit-team / junit4](#)

 Wiki

“

Verifying that code completes normally is important, but making sure the code behaves as expected in exceptional situations is vital too.

”



# stackoverflow

How do you assert that  
a certain exception is  
thrown in JUnit 4 tests?

12 years old

1.4 million views

highly active

## @Test(expected)

---

```
@Test(expected = Throwable.class)
public void exceptionalTest() {
    throwingCode();
}
```

## assertThrows

---

```
@Test
public void exceptionalTest() {
    assertThrows(
        Throwable.class,
        () -> throwingCode()
    );
}
```

## try/catch

---

```
@Test
public void exceptionalTest() {
    try {
        throwingCode();
        fail();
    } catch (Throwable t) {
        // expected
    }
}
```

## @Rule

---

```
@Rule
public ExpectedException thrown =
    ExpectedException.none();

@Test
public void exceptionalTest() {
    thrown.expect(Throwable.class);
    throwingCode();
}
```

## generic

---

```
@Test
public void exceptionalTest() {
    assertThatThrownBy(
        () -> throwingCode()
    ).isInstanceOf(Throwable.class);
}
```

# Exceptional Tests

An **Exceptional Test** is a test that exercises exceptional behavior

# Investigation of Exceptional Tests



Do developers **write** exceptional tests?



What **kinds** of exceptions do they test?



# Applications



**reveal** information on  
thrown exceptions



**improve** documentation

# Analyzed Projects and Tests

1 157

Java **projects**  
on GitHub

1.1 M

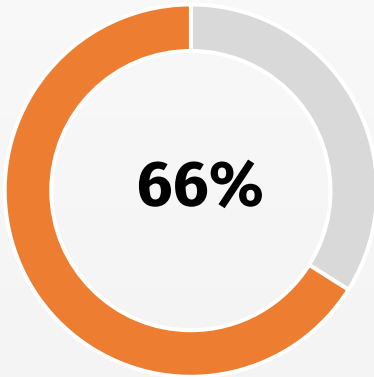
JUnit  
**tests**

14 M

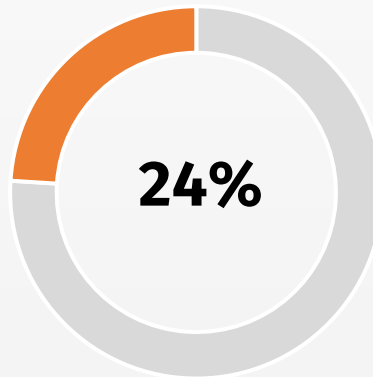
**lines** of  
test code

# How Many Exceptional Tests?

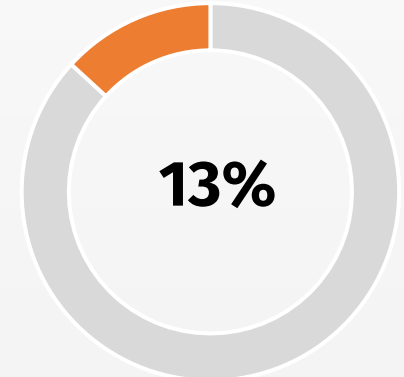
Projects



Test Classes



Test Methods



# Origin of the Exceptions

■ Java    ■ External    ■ Local

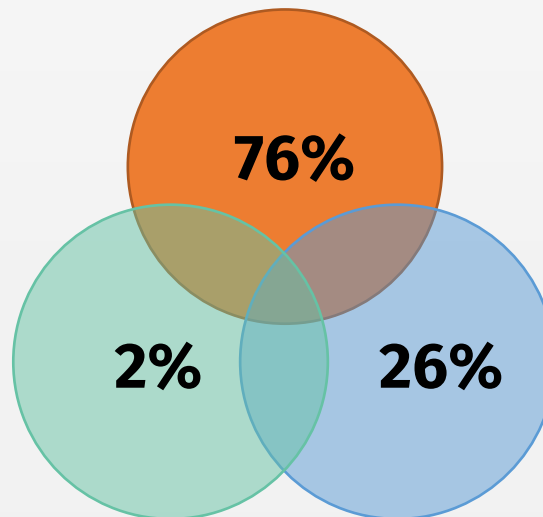


# Prevalence of Java Exceptions

■ Java   ■ External   ■ Local



## Exceptional Tests



# Kind of the Exceptions

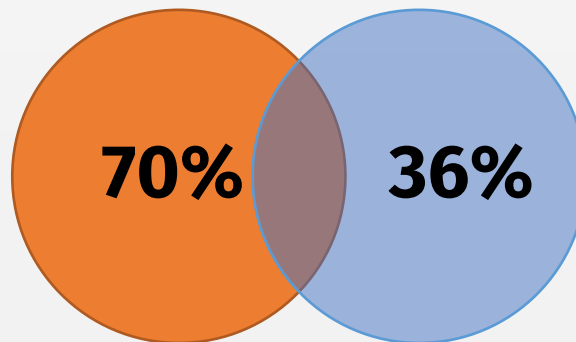
■ Unchecked    ■ Checked



# Prevalence of Unchecked Exceptions



## Exceptional Tests





Tests  
targeting  
unchecked  
exceptions  
may **reveal**  
helpful  
information

```
// Set xattr with null name
try {
    fs.setXAttr(path, null, value1, EnumSet.of(XAttrSetFlag.CREATE,
        XAttrSetFlag.REPLACE));
    Assert.fail("Setting xattr with null name should fail.");
} catch (NullPointerException e) {
    GenericTestUtils.assertExceptionContains("XAttr name cannot be null", e);
} catch (RemoteException e) {
    GenericTestUtils.assertExceptionContains("Required param xattr.name for "
        + "op: SETXATTR is null or empty", e);
}
```





# Exceptional tests may improve documentation

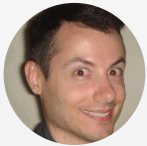
```
// Set xattr with null name
try {
    fs.setXAttr(path, null, value1, EnumSet.of(XAttrSetFlag.CREATE,
        XAttrSetFlag.REPLACE));
    Assert.fail("Setting xattr with null name should fail.");
} catch (NullPointerException e) {
    GenericTestUtils.assertExceptionContains("XAttr name cannot be null", e);
} catch (RemoteException e) {
    GenericTestUtils.assertExceptionContains("Required param xattr.name for "
        + "op: SETXATTR is null or empty", e);
}

/**
 * Set an xattr of a file or directory.
 * The name must be prefixed with the namespace followed by ".". For example,
 * "user.attr".
 * <p>
 * Refer to the HDFS extended attributes user documentation for details.
 *
 * @param path Path to modify
 * @param name xattr name.
 * @param value xattr value.
 * @throws IOException IO failure
 * @throws UnsupportedOperationException if the operation is unsupported
 *         (default outcome).
 */
public void setXAttr(Path path, String name, byte[] value)
```

# How Java Programmers Test Exceptional Behavior



Diego Marcilio



Carlo A. Furia



Università  
della  
Svizzera  
italiana

Software  
Institute



<https://dvmarcilio.github.io/msr21/>