

# Jester

- Java bytecode manipulator for test verification
- Modifies method under test changing branching logic, return values, etc.
- If your tests still pass after modification, you're obviously missing a test.



```
Use classpath: src2/java/main:../jester136/jester.jar
:../jester136:target/lib/junit-3.8.1.jar:target/lib/dom4j-core-1.4-dev-8.jar
:target/lib/jdom-b10.jar:target/lib/jdom-b10.jar:target/lib/xom-1.0d21.jar
:target/test-classes:target/classes
```

```
...
src2/java/main/org/jaxen/BaseXPath.java
- changed source on line 192 (char index=7757) from 1 to 2
    answer.size() == ?1 )
    {
        Object first = answ
```

```
src2/java/main/org/jaxen/BaseXPath.java
- changed source on line 691 (char index=24848) from 0 to 1

        return results.get( ?0 );
    }
}
```

lots more output...

```
src2/java/main/org/jaxen/BaseXPath.java
- changed source on line 691 (char index=24848) from 0 to 1

        return results.get( ?0 );
    }
}
```

```
10 mutations survived out of 11 changes. Score = 10
took 1 minutes
```

```
def process_defn(exp)
  self.method = exp.shift
  result = [:defn, method]

  result << process(exp.shift) until exp.empty?

  jester(result) if should_jester?

  return result
end

def process_if(exp)
  cond = process(exp.shift)
  t = process(exp.shift)
  f = process(exp.shift)

  if should_jester? then
    [:if, cond, f, t]
  else
    [:if, cond, t, f]
  end
end
```

```
def many? n
  if n >= 3 then
    true
  else
    false
  end
end
```



```
% ruby -Ilib bin/jester test/test_jester.rb TestJester many?
Validating test/test_jester.rb
loaded
Loaded suite bin/jester
Started
.
Finished in 0.000224 seconds.

1 tests, 0 assertions, 0 failures, 0 errors
Tests passed -- jestering
Jestering TestJester#many?

Replacing #many? with:
def many?(n)
  if (n >= 3) then
    false
  else
    true
  end
end
end

(eval):1: warning: method redefined; discarding old many?
Loaded suite bin/jester
Started
.
Finished in 0.000227 seconds.

1 tests, 0 assertions, 0 failures, 0 errors

*** Tests passed again after jestering, your tests are incomplete
```

```
% ruby -Ilib bin/jester test/test_jester.rb TestJester many?
```

```
Validating test/test_jester.rb
```

```
loaded
```

```
Loaded suite bin/jester
```

```
Started
```

```
.  
Finished in 0.00029 seconds.
```

```
1 tests, 3 assertions, 0 failures, 0 errors
```

```
Tests passed -- jestering
```

```
Jestering TestJester#many?
```

```
Replacing #many? with:
```

```
def many?(n)  
  if (n >= 3) then  
    false  
  else  
    true  
  end  
end
```

```
(eval):1: warning: method redefined; discarding old many?
```

```
Loaded suite bin/jester
```

```
Started
```

```
F
```

```
Finished in 0.00692 seconds.
```

```
1) Failure:
```

```
test_many_eh_good(TestJester) [./test/test_jester.rb:22]:
```

```
<false> is not true.
```

```
1 tests, 1 assertions, 1 failures, 0 errors
```

```
Tests failed -- this is good
```