

Let's just test that

real quick

with **Jan Stępień**

@janstepien jan@stepien.cc



@innoQ



@cljmuc

I want *you* to become

better at testing



time/date

GARMIN

training

Today
6:18

11.83^{km/h}
836^{rpm}

01:03:17
11.2^{km/h}

GPS

menu

lap / reset

```
double stepsPerHour(int steps, double seconds) {  
    return 60 * 60 * steps / seconds;  
}
```

```
assertThat(stepsPerHour(1, 2), equalTo(0.5));  
assertThat(stepsPerHour(2, 2), equalTo(1));  
assertThat(stepsPerHour(0, 3), equalTo(0));
```

```
@Test(expected = ArithmeticException.class)  
public void testZeroSeconds() {  
    stepsPerHour(1, 0.0);  
}
```

```
[(1, 2, 0.5),  
 (2, 2, 1),  
 (0, 3, 0)].stream().map((steps, secs, value) → {  
     assertThat(stepsPerHour(steps, secs),  
                 equalTo(value));  
});
```

$$\frac{a}{b} = \frac{a}{b}$$

$$\frac{a}{b} = \frac{a}{b}$$

$$\frac{a}{b} \cdot b = \frac{a}{b} \cdot b$$

$$\frac{a}{b} = \frac{a}{b}$$

$$\frac{a}{b} \cdot b = \frac{a}{b} \cdot b$$

$$\frac{a}{b} \cdot b = a$$

```
[(1, 2, 0.5),  
 (2, 2, 1),  
 (0, 3, 0)].stream().map((steps, secs, value) → {  
     assertThat(stepsPerHour(steps, secs),  
                 equalTo(value));  
});
```

```
[(1, 2),  
 (2, 2),  
 (0, 3)].stream().map((steps, secs) → {  
     assertThat(3600 * secs * stepsPerHour(steps, secs),  
                 equalTo(steps));  
});
```

QuickCheck

junit-quickcheck

github.com/pholser/junit-quickcheck

```
@RunWith(JUnitQuickcheck.class)
public class StepsPerSecondProperties {
    @Property
    void stepsProperty(int steps, double secs) {
        assertEquals(3600 * secs * stepsPerHour(steps, secs),
            steps);
    }
}
```

```
int[] reverse(int[] array) {  
    ...  
}
```




```
@RunWith(JUnitQuickcheck.class)
public class ReverseProperties {
    @Property
    void reverseProperty(int[] array) {
        ...
    }
}
```

```
@RunWith(JUnitQuickcheck.class)
public class ReverseProperties {
    @Property
    void reverseProperty(int[] array) {
        assertEquals(reverse(reverse(array)),
            array);
    }
}
```

```
{
  name: "Orson Welles",
  cinematography: [
    {
      title: "Citizen Kane",
      year: 1941,
      director: { name: "Orson Welles" },
      screenwriter: { name: "Orson Welles" },
      starring: [
        { name: "Orson Welles" },
        { name: "Joseph Cotten" },
        { name: "Dorothy Comingore" }
      ]
    }
  ]
}
```

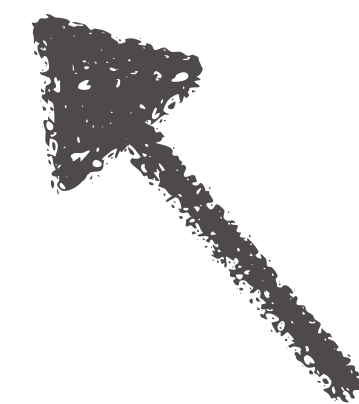


GET /hosen/lee/farbe-hellblau/40-70-euro/seite-2

erty-based testing

erty-based testing

tice with **Jan Stępień**



wat

GET /hosen/lee/farbe-hellblau/40-70-euro/seite-2

State happens

RedisCache

RedisCache

accompanied by an oracle

RedisCache

accompanied by an oracle

java.util.HashMap

```
[“put”, “foo”, “bar”]  
[“clear”]  
[“put”, “foo”, “123”]  
[“get”, “foo”]  
[“count”]
```

*Testing the Hard Stuff
and Staying Sane*

John Hughes

[α β γ δ]
[1 2 3 4]

[α β γ δ]

[1 2 3 4]

[α β γ δ 1 2 3 4]

[[[α β γ δ]
[1 2 3 4]
[α β 1 2 3 4 γ δ]

[[[α β γ δ]
[1 2 3 4]
[α 1 β 2 γ 3 δ 4]

Jepsen: On the perils of network partitions

Kyle “Aphyr” Kingsbury

I want *you* to become

better at testing

Let's just test that

real quick

with **Jan Stępień**

@janstepien jan@stepien.cc