

Title

Tool

By

Generating Test Data for Fun and Profit

Featuring: The Factorial Function!

Title

Generating Test Data for Fun and Profit

Featuring: The Factorial Function!

Tool

ScalaCheck

<http://code.google.com/p/scalacheck/>
Works on both Scala and Java code!

By

Title

Generating Test Data for Fun and Profit

Featuring: The Factorial Function!

Tool

ScalaCheck

<http://code.google.com/p/scalacheck/>
Works on both Scala and Java code!

By

Mitch Blevins

<http://cleverlytitled.blogspot.com/>

Out

Test

Code

```
def fac(i: Int): Int = {  
    i * fac(i-1)  
}
```

Code

```
def fac(i: Int): Int = {  
  i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forall(choose(1,10)) {  
    i: Int =>  
      fac(i) > 0  
  }
```

Out

Code

```
def fac(i: Int): Int = {  
  i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(1,10)) {  
    i: Int =>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive: i=1  
> java.lang.StackOverflowError
```

Code

```
def fac(i: Int): Int = {  
  if (i == 0) 1  
  else i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(1,10)) {  
    i: Int =>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive:  
> OK, passed 100 tests.
```

Code

```
def fac(i: Int): Int = {  
  if (i == 0) 1  
  else i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(-10, 10)) {  
    i: Int =>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive: i=-1  
> java.lang.StackOverflowError
```


Code

```
def fac(i: Int): Int = {  
  if (i < 0) throw new IAE("pos")  
  if (i == 0) 1  
  else i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(-10,10)) {  
    i: Int => (i >= 0) ==>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive:  
> OK, passed 100 tests.
```

Code

```
def fac(i: Int): Int = {  
  if (i < 0) throw new IAE("pos")  
  if (i == 0) 1  
  else i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(-10, 100)) {  
    i: Int => (i >= 0) ==>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive: i=17  
> Falsified (-288522240)
```

Code

```
def fac(i: Int): Long = {  
  if (i < 0) throw new IAE("pos")  
  if (i == 0) 1  
  else i * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(-10,100)) {  
    i: Int => (i>=0) ==>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive: i=21  
> Falsified (-4249290049419214848)
```

Code

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  if (i == 0) BigInt(1)  
  else BigInt(i) * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(-10,100)) {  
    i: Int => (i>=0) ==>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive:  
> OK, passed 100 tests.
```

Code

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  if (i == 0) BigInt(1)  
  else BigInt(i) * fac(i-1)  
}
```

Test

```
property("positive") =  
  forAll(choose(-10, 10000)) {  
    i: Int => (i >= 0) ==>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive: i=5347  
> java.lang.StackOverflowError
```

Code

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  var (n, acc) = (1, BigInt(1))  
  while (n < i) { n+=1; acc*=n }  
  acc  
}
```

Test

```
property("positive") =  
  forAll(choose(-10, 10000)) {  
    i: Int => (i >= 0) ==>  
      fac(i) > 0  
  }
```

Out

```
Factorial.positive:  
> OK, passed 100 tests.
```

Code

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  var (n, acc) = (1, BigInt(1))  
  while (n < i) { n+=1; acc*=n }  
  acc  
}
```

Test

```
property("positive") =  
  forAll(choose(-10, 1000000)) {  
    i: Int => (i >= 0) ==>  
      fac(i) > 0  
  }
```

Out

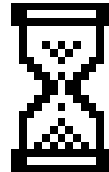
```
Factorial.positive:  
> OK, passed 100 tests.
```

Code

Test

Out

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  var (n, acc) = (1, BigInt(1))  
  while (n < i) { n+=1; acc*=n }  
  acc  
}  
property("positive") =  
  forall {  
    i: Int => (i >= 0) ==>  
      fac(i) > 0  
  }
```



Code

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  var (n, acc) = (1, BigInt(1))  
  while (n < i) { n+=1; acc*=n }  
  acc  
}
```

Test

```
property("increasing") =  
  forAll(choose(-10, 1000000)) {  
    i: Int => (i > 1) ==>  
      fac(i) > fac(i-1)  
  }
```

Out

```
Factorial.increasing:  
> OK, passed 100 tests.
```

Code

```
def fac(i: Int): BigInt = {  
  if (i < 0) throw new IAE("pos")  
  var (n, acc) = (1, BigInt(1))  
  while (n < i) { n+=1; acc*=n }  
  acc  
}
```

Test

```
property("matches check") =  
  forAll(choose(-10, 1000000)) {  
    i: Int => (i >= 0) ==>  
      (fac(i) > checkFactorial(i))  
  }
```

Out

```
Factorial.matches check:  
> OK, passed 100 tests.
```



**My
Death
Star
is
Inpenetrable**

(why test?)



**Why didn't I test
thermal exhaust
port #38634?**

**I had friends on
that Death Star.**