1. Introduction

This document contains the example unit test to prove that unit tests are working for our language. We selected Python as development language, so the files are python files. We translated the code of the example unit test from http://wiki.fernuni-hagen.de/eclipse/index.php/Einf%C3%BChrung_in_JUnit to Python. As the check for infinite loops without a crash of the whole unit test is a lot more complicated in Python than in Java, the check for the square root function was not executed. If you remove the comment marks, the unit test will crash, so you are aware that there is an infinite loop.

In section 2 you can find the failing test examples and in section 3 the corrected examples are shown.

2. Non-Working Unit Test

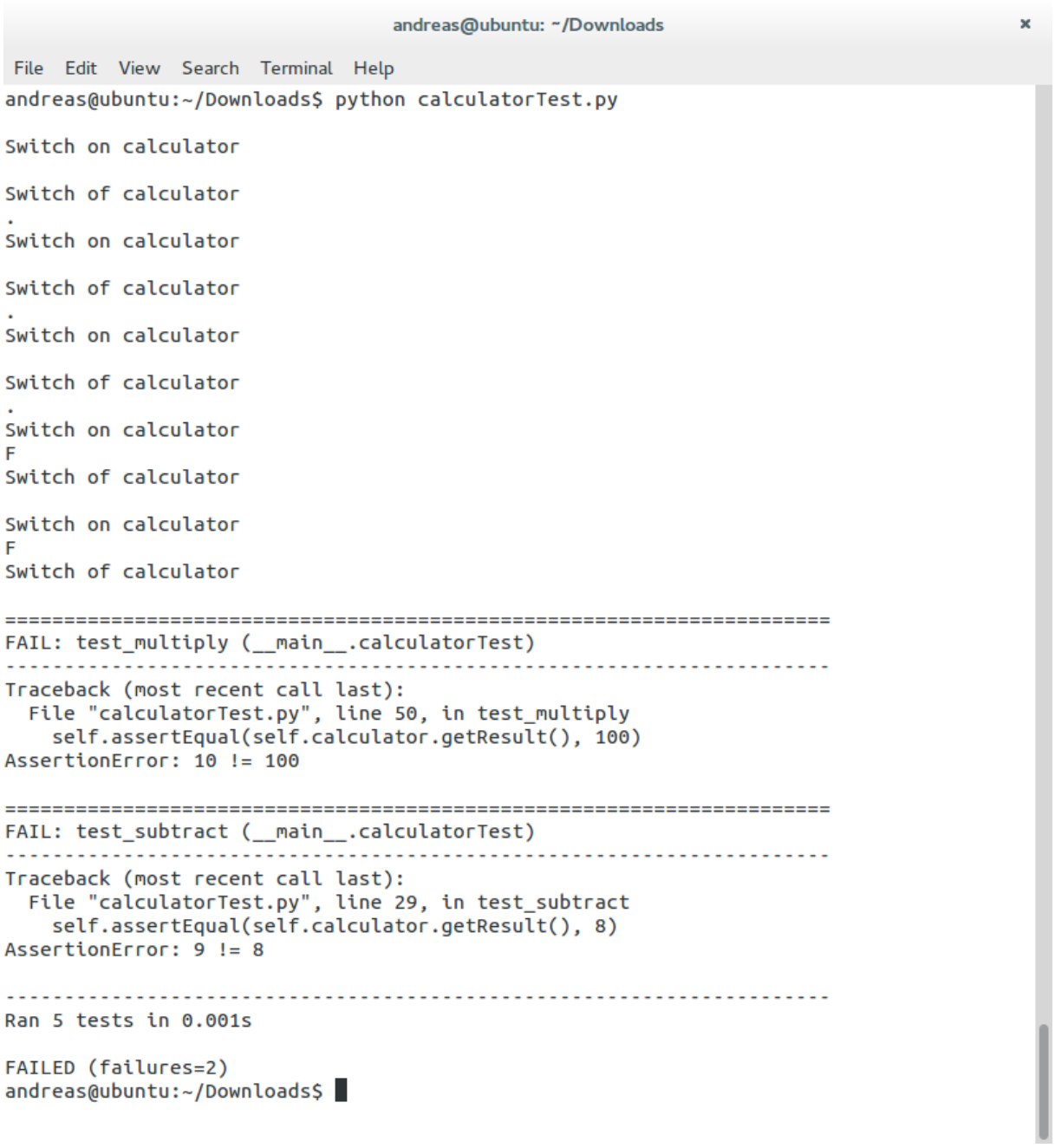
The calculator file (calculator.py):

```
class Calculator(object):

	def __init__(self):
		self.result = 0

	def add(self, n):
		self.result = self.result + n

	def subtract(self, n):
		self.result = self.result - 1

	def multiply(self, n):
		pass

	def divide(self, n):
		self.result = self.result / n

	def square(self, n):
		self.result = n * n

	def squareRoot(self, n):
		while True:
			pass

	def clear(self):
		self.result = 0

	def switchOn(self):
		self.result = 0

	def switchOff(self):
		pass

	def getResult(self):
		return self.result
```

The unit test file (calculatorTest.py):

```
import calculator
import unittest

class calculatorTest(unittest.TestCase):

	def setUp(self):
		print "\nSwitch on calculator"
		self.calculator = calculator.Calculator()
		self.calculator.switchOn()
		self.calculator.clear()

	def tearDown(self):
		print "\nSwitch of calculator"
		self.calculator.switchOff()

	def test_add(self):
		self.calculator.add(1)
		self.calculator.add(1)
		self.assertEqual(self.calculator.getResult(), 2)

	def test_subtract(self):
		self.calculator.add(10)
		self.calculator.subtract(2)
		self.assertEqual(self.calculator.getResult(), 8)

	def test_divide(self):
		self.calculator.add(8)
		self.calculator.divide(2)
		self.assertEqual(self.calculator.getResult(), 4)

	def test_divideByZero(self):
		self.assertRaises(ZeroDivisionError, self.calculator.divide, 0)

	#def test_squareRoot(self):
	#	self.calculator.squareRoot(100)
	#	self.assertEqual(calculator.getResult(), 10)

	def test_multiply(self):
		self.calculator.add(10)
		self.calculator.multiply(10)
		self.assertEqual(self.calculator.getResult(), 100)

if __name__ == '__main__':
	unittest.main()
```

Test result:


andreas@ubuntu: ~/Downloads
File Edit View Search Terminal Help
andreas@ubuntu:~/Downloads$ python calculatorTest.py
Switch on calculator
Switch of calculator
.
Switch on calculator
Switch of calculator
.
Switch on calculator
Switch of calculator
.
Switch on calculator
F
Switch of calculator
Switch on calculator
F
Switch of calculator
FAIL: test_multiply (__main__.calculatorTest)
Traceback (most recent call last):
File "calculatorTest.py", line 50, in test_multiply
self.assertEqual(self.calculator.getResult(), 100)
AssertionError: 10 != 100
FAIL: test_subtract (__main__.calculatorTest)
Traceback (most recent call last):
File "calculatorTest.py", line 29, in test_subtract
self.assertEqual(self.calculator.getResult(), 8)
AssertionError: 9 != 8
Ran 5 tests in 0.001s
FAILED (failures=2)
andreas@ubuntu:~/Downloads$

3. Working Unit Test

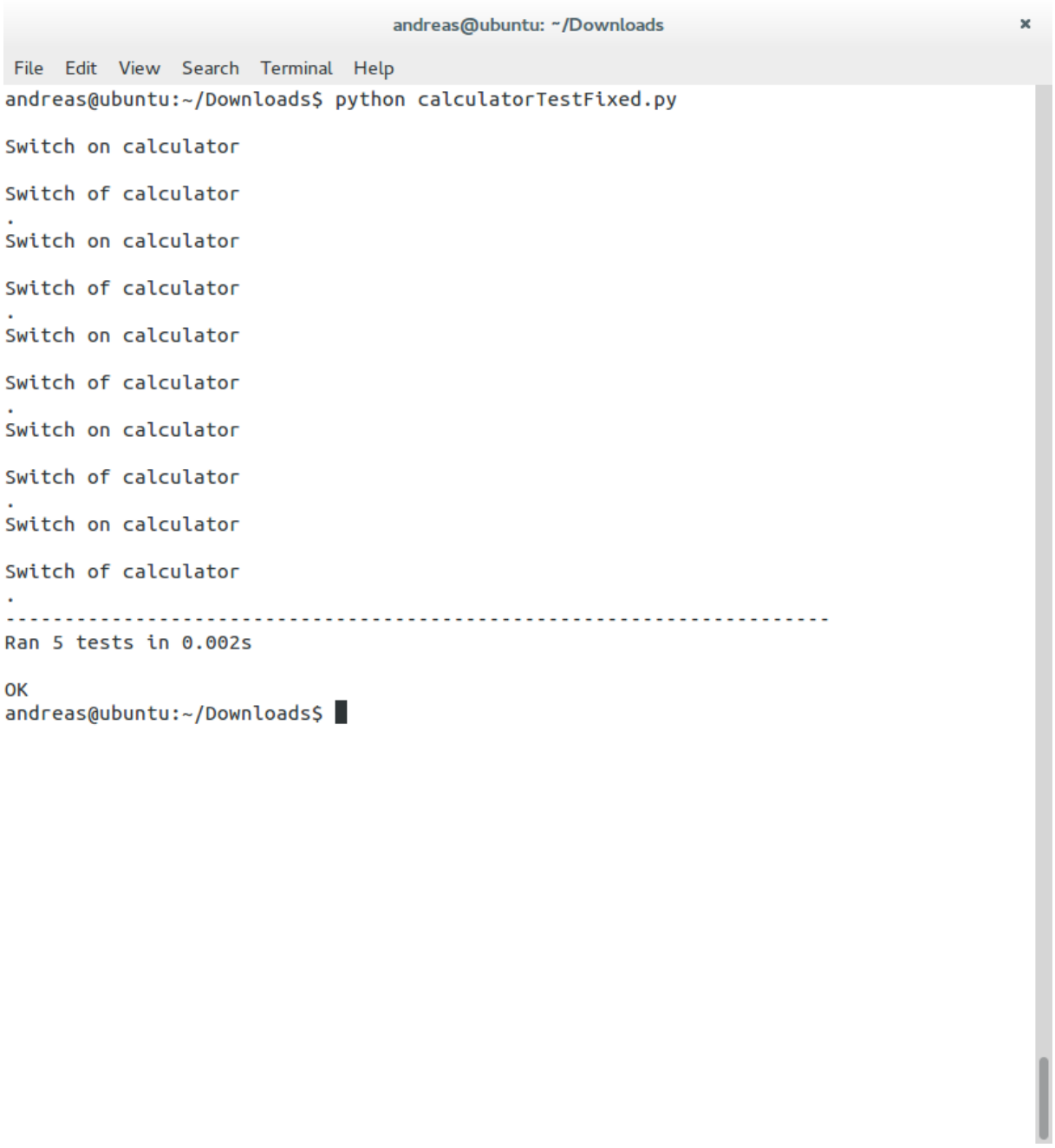
The calculator file (calculator.py) is corrected:

```
class Calculator(object):

	def __init__(self):
		self.result = 0

	def add(self, n):
		self.result = self.result + n

	def subtract(self, n):
		self.result = self.result - n

	def multiply(self, n):
		self.result = self.result * n

	def divide(self, n):
		self.result = self.result / n

	def square(self, n):
		self.result = n * n

	def squareRoot(self, n):
		while True:
			pass

	def clear(self):
		self.result = 0

	def switchOn(self):
		self.result = 0

	def switchOff(self):
		pass

	def getResult(self):
		return self.result
```

The unit test file (calculatorTest.py) is unchanged:

```
import calculator
import unittest

class calculatorTest(unittest.TestCase):

	def setUp(self):
		print "\nSwitch on calculator"
		self.calculator = calculator.Calculator()
		self.calculator.switchOn()
		self.calculator.clear()

	def tearDown(self):
		print "\nSwitch of calculator"
		self.calculator.switchOff()

	def test_add(self):
		self.calculator.add(1)
		self.calculator.add(1)
		self.assertEqual(self.calculator.getResult(), 2)

	def test_subtract(self):
		self.calculator.add(10)
		self.calculator.subtract(2)
		self.assertEqual(self.calculator.getResult(), 8)

	def test_divide(self):
		self.calculator.add(8)
		self.calculator.divide(2)
		self.assertEqual(self.calculator.getResult(), 4)

	def test_divideByZero(self):
		self.assertRaises(ZeroDivisionError, self.calculator.divide, 0)

	#def test_squareRoot(self):
	#	self.calculator.squareRoot(100)
	#	self.assertEqual(calculator.getResult(), 10)

	def test_multiply(self):
		self.calculator.add(10)
		self.calculator.multiply(10)
		self.assertEqual(self.calculator.getResult(), 100)

if __name__ == '__main__':
	unittest.main()
```

Test result:


andreas@ubuntu: ~/Downloads
File Edit View Search Terminal Help
andreas@ubuntu:~/Downloads$ python calculatorTestFixed.py
Switch on calculator
Switch of calculator
.
Switch on calculator
Switch of calculator
.
Switch on calculator
Switch of calculator
.
Switch on calculator
Switch of calculator
.
Switch on calculator
Switch of calculator
.

Ran 5 tests in 0.002s
OK
andreas@ubuntu:~/Downloads$