Python - Teil 6

Michael Möbius

Agenda

- Properties
- Unit Tests
- Code coverage

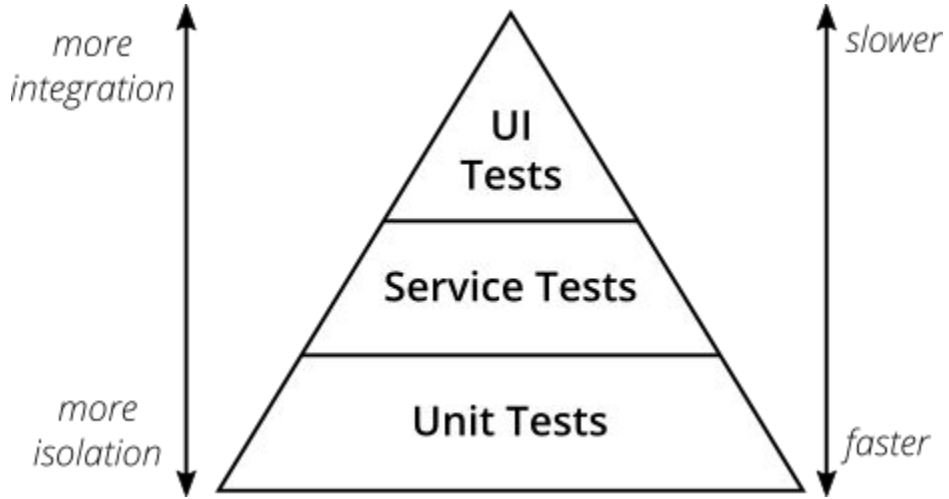
Getter/Setter oder öffentliche Variablen

- Getter/Setter wenn zugriff geschützt werden soll
- Bei einfachen Variablen auch öffentlich
  - Und nachträglich als Property falls Zugriffsschutz nötigt

Properties

```
31  class P3:
32
33      def __init__(self, x):
34          self.x = x
35
36      @property
37      def x(self):
38          return self.__x
39
40      @x.setter
41      def x(self, x):
42          if x < 0:
43              self.__x = 0
44          elif x > 1000:
45              self.__x = 1000
46          else:
47              self.__x = x
48
49  p3 = P3(42)
50  print(p3.x)
51  p3.x = 47
52  print(p3.x)
```

Test Pyramide



Unit Tests

- Automatisierte Tests für den Code
- Klasse ableiten von “unittest.TestCase”
- Methoden mit “def test_***”
- def setUp(self)
- def tearDown(self)
- python -m unittest discover
- python -m unittest test_module.TestClass

Testing Std in/out

```
@contextmanager
def captured_std():
    new_out, new_err, new_in = StringIO(), StringIO(), StringIO()
    old_out, old_err, old_in = sys.stdout, sys.stderr, sys.stdin
    try:
        sys.stdout, sys.stderr, sys.stdin = new_out, new_err, new_in
        yield sys.stdout, sys.stderr, sys.stdin
    finally:
        sys.stdout, sys.stderr, sys.stdin = old_out, old_err, old_in

class stdTest(unittest.TestCase):

    def testPrint(self):
        with captured_std() as (out, err, inp):
            Std().print123()
        # This can go inside or outside the `with` block
        output = out.getvalue().strip()
        self.assertEqual(output, '123')
```

Coverage

- **Aufzeigen der Testabdeckung**
- pip install coverage
- coverage run -m unittest discover
- coverage html