МОДУЛЬНЕ ТЕСТУВАННЯ

Фреймворки для автономного тестування

- **unittest** (стандартна бібліотека, у стилі Java, C#);
- **nose** (для “швидкого” тестування;
- **pytest** (потужний, у пайтонівському стилі).

unittest

```python
# funcs.py

def uniques(text):
    return len(set(text))

def mean(*args):
    return sum(args)/len(args)
```

```
..
-----------------------------
Ran 2 tests in 0.042s

OK
```

```python
# test_funcs.py

import unittest
from funcs import *

class FuncsTest(unittest.TestCase):

    def test_mean(self):
        self.assertEqual(mean(1, 2, 9), 4)

    def test_uniques(self):
        self.assertEqual(uniques('txt'), 2)

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

Методи перевірок

```
assertEqual(a, b) — a == b
assertNotEqual(a, b) — a != b
assertTrue(x) — bool(x) is True
assertFalse(x) — bool(x) is False
assertIs(a, b) — a is b
assertIsNone(x) — x is None
assertIn(a, b) — a in b
assertNotIn(a, b) — a not in b
assertRaises(exc, fun, *args, **kwds) — fun викликає виняток exc
assertGreater(a, b) — a > b
assertListEqual(a, b)
...
```

Інтерфейс командного рядка (CLI)

```
> python3 -m unittest test_funcs.py
..
----------------------------------------------------------------------
Ran 2 tests in 0.000s
OK
> python3 -m unittest test_funcs.FuncsTest
..
----------------------------------------------------------------------
Ran 2 tests in 0.001s
OK
> python3 -m unittest test_funcs.FuncsTest.test_mean
.
----------------------------------------------------------------------
Ran 1 test in 0.000s
OK
```

CLI: детальніша інформація

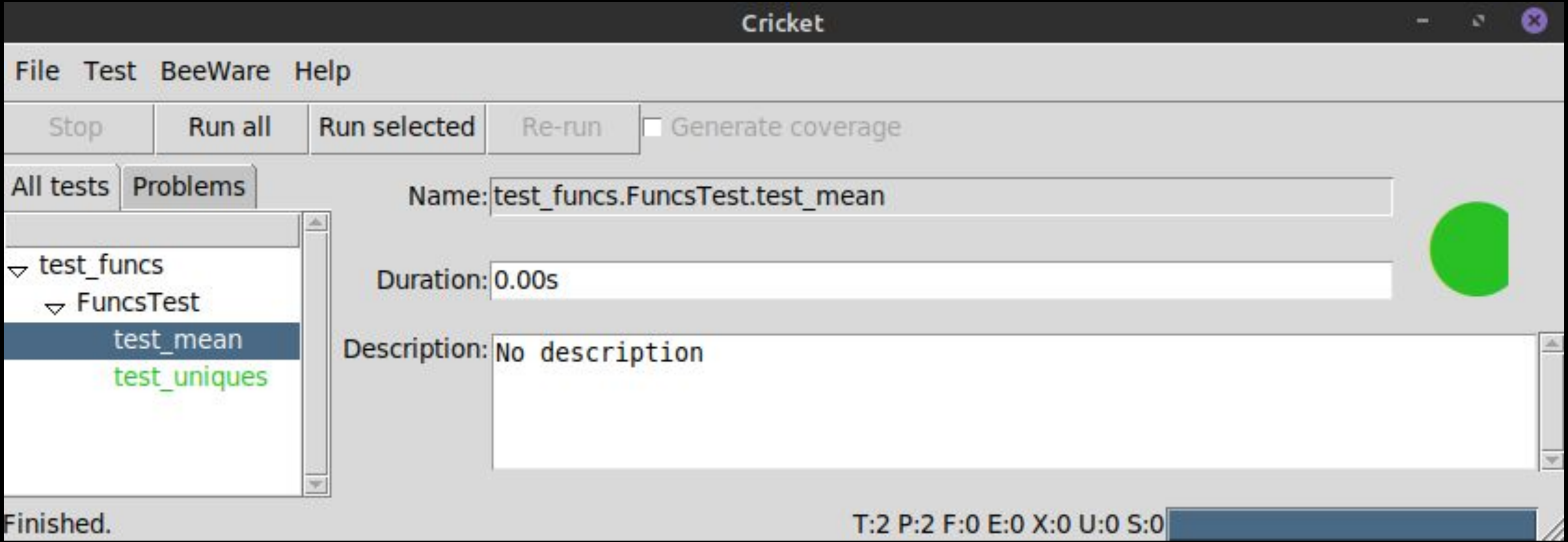
```
> python3 -m unittest -v test_funcs.py
test_mean (test_funcs.FuncsTest) ... ok
test_uniques (test_funcs.FuncsTest) ... ok

----------------------------------------------------------------------
Ran 2 tests in 0.000s

OK
```

CLI: Test Discovery

```
> python3 -m unittest -v
test_mean (test_funcs.FuncsTest) ... ok
test_uniques (test_funcs.FuncsTest) ... ok

----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK
```

Графічний інтерфейс

```
> pip3 install cricket

> cricket-unittest
```



Name: test_funcs.FuncsTest.test_mean

Duration: 0.00s

Description: No description

Finished.

T:2 P:2 F:0 E:0 X:0 U:0 S:0

setUpClass / tearDownClass

```python
class FuncsTest(unittest.TestCase):

   @classmethod
   def setUpClass(cls):
      print('---START---')

   @classmethod
   def tearDownClass(cls):
      print('---FINISH---')

   def test_mean(self):
      """TEST-1"""
      ...

   ...
```

```
> python3 -m unittest -v
---START---
test_mean (test_funcs.FuncsTest)
TEST-1 ... ok
test_uniques (test_funcs.FuncsTest)
TEST-2 ... ok
---FINISH---

------------------------------------
-
Ran 2 tests in 0.000s

OK
```

setUp / tearDown

```
class FuncsTest(unittest.TestCase):

   ...

   def setUp(self):
      print(f'\nStart
            {self.shortDescription()}')

   def tearDown(self):
      print(f'Finish
            {self.shortDescription()}')

   ...
```

```
> python3 -m unittest -v
---START---
test_mean (test_funcs.FuncsTest)
TEST-1 ...
Start TEST-1
Finish TEST-1
ok
test_uniques (test_funcs.FuncsTest)
TEST-2 ...
Start TEST-2
Finish TEST-2
ok
---FINISH---
-----------------------------------
Ran 2 tests in 0.001s
OK
```

Безумовне пропускання тесту

```
class FuncsTest(unittest.TestCase):

   @unittest.skip("Reason: some reason")
   def test_mean(self):
       self.assertEqual(mean(1, 2, 9), 4)
```

```
> python3 -m unittest
test_mean (test_funcs.FuncsTest) ... skipped 'Reason: some reason'
test_uniques (test_funcs.FuncsTest) ... ok
----------------------------------------------------------------------
Ran 2 tests in 0.001s
OK (skipped=1)
s.
----------------------------------------------------------------------
Ran 2 tests in 0.001s
OK (skipped=1)
```

Провал тесту

```
> python3 -m unittest
---START---
Start TEST-1
Finish TEST-1
F
Start TEST-2
Finish TEST-2
.---FINISH---
======================================================================
FAIL: test_mean (test_funcs.FuncsTest)
TEST-1
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/home/mokasin/Документи/Дисципліни/code/unittest/test_funcs.py", line 22, in
test_mean
    self.assertEqual(mean(1, 2, 9), 4)
AssertionError: 12 != 4
----------------------------------------------------------------------
Ran 2 tests in 0.001s
FAILED (failures=1)
```

Організація тестування

```
# funcs.py

def uniques(text):
    return len(set(text))

def mean(*args):
    return sum(args)/len(args)

def is_positive(x):
    return x > 0
```

```
# test_funcs.py

import unittest
from funcs import *

class FuncsTest(unittest.TestCase):
   ...

class OtherTest(unittest.TestCase):

   def test_is_positive(self):
       self.assertTrue(is_positive(10))
       self.assertFalse(is_positive(0))

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

TestSuite

```
# test_runner.py

import unittest
import test_funcs
test = unittest.TestSuite()
test.addTest(unittest.makeSuite(test_funcs.FuncsTest))
test.addTest(unittest.makeSuite(test_funcs.OtherTest))
runner = unittest.TextTestRunner(verbosity=2)
runner.run(test)
```

```
> python3 test_runner.py
test_mean (test_funcs.FuncsTest) ... ok
test_uniques (test_funcs.FuncsTest) ... ok
test_is_positive (test_funcs.OtherTest) ... ok
----------------------------------------------------------------------
Ran 3 tests in 0.000s
OK
```