Programming Lab

Lezione 7

Testing e unit tests

Stefano Alberto Russo

Cos’è il testing

- Per testing si intende il testare, in genere in modo automatico, delle cose. Dal software, ad una penna, al vostro cellulare.

- Il testing del software è più facile di quello dell’hardware (dove servono attrezzature specializzate), perchè basta scrivere del software in più

- Esiste chi fa il testing del software di testing :)

Cos’è il testing

Pseudocodice per un generico concetto di testing:

dato un **input** e un **output** *noto*

input *noto* → CODICE → **output**

if **output** != **output** *noto*:

    errore!

Cos’è il testing

Codice Python per il testing di una funzione che fa la somma di due numeri:

```
# Funzione somma
def somma(a,b):
    return a+b

# Testing
if not somma(1,1) == 2:
    raise Exception('Test 1+1 non passato')

if not somma(1.5,2.5) == 4:
    raise Exception('Test 1.5+2.5 non passato')
```

Testing vs unit testing

Il testing generico può anche essere effettuato su tutto il codice /programma.

input → **programma o funzione molto grossa** → output

Se invece testo le “minime” unità testabili, allora si parla di *unit testing*:

input → **funzione piccola** → output
input → **oggetto piccolo**→ output
input → **altra funzione piccola**→ output

In questo modo sono molto più granulare nel capire dove è andato storto cosa.

Il modulo unittest

lezione7.py

```
# Funzione somma
def somma(a,b):
    return a+b
```

test_lezione7.py

```
import unittest
from lezione7 import somma

# Testing
class TestSomma(unittest.TestCase):

    def test_somma(self):
        self.assertEqual(somma(1,1), 2)
        self.assertEqual(somma(1.5,2.5), 4)
```

```
~/ProgrammingLab2021$ python -m unittest discover
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
```

Come verrete valutati

Il vostro esame verrà valutato con degli unit test. Esempio con l'oggetto CSVFile:

```
import unittest
from esame import CSVFile

class TestCSVFile(unittest.TestCase):

    def test_init(self):

        csv_file = CSVFile('shampoo_sales.csv')

        # Controllo che il nome del file sia stato salvato
        # in un attributo dell'oggetto di nome "name"
        self.assertEqual(csv_file.name, 'shampo_sales.csv')
```

P.s. sviluppare coide test-driven

Un bellissimo modo per sviluppare codice è essere test-driven:

PRIMA scrivo i test, POI il codice.

In questo modo mi focalizzo prima su che cosa voglio che faccia il codice, e se sono bravo prevedo anche i casi strani (se passo una stringa alla funzione somma cosa voglio che succeda?)

..non è richiesto per questo corso, è giusto un accenno per voi.

Esercizio

Scrivete dei test o unit test (a seconda di cosa più vi aggrada) per gli oggetti **CSVFile** e **NumericalCSVFile**.

Potete per esempio:

- creare vari file di test e verificare che la funzione get data dia sempre l'output che vi aspettate
- verificare che il nome del file sia salvato come attributo
- verificare che vengano alzate specifiche eccezioni

  (usare il costrutto *try-except* o *self.assertRaises()* se si usa il modulo unittest)