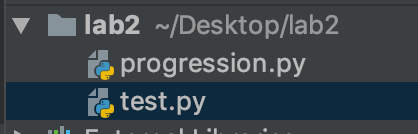
Лабораторна робота №2 з предмету «Технології програмування»

Виконала студентка 3го курсу Баранюк Катерина

Тестування програмного забезпечення

Мета: розглянути принципи модульного тестування програмного забезпечення.

Хід роботи:



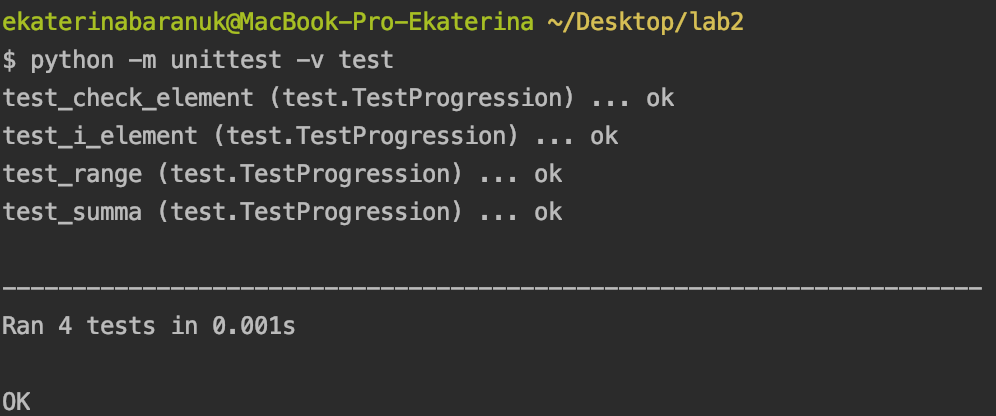
Створення двух файлів: progression.py-файл з безпосередньо кодом та test.py файл з кодом завдяки якому ми проводимо тестування.

Код програми:

class Progression:  
 def \_\_init\_\_(self, a, d, n):  
 self.a = a # first element  
 self.d = d # difference  
 self.n = n # amount of elements  
  
 def i\_element(self, j):  
 aj = self.a + (j - 1) \* self.d  
 return aj  
  
 def summa(self, k):  
 s = (k \* (2 \* self.a + self.d \* (k - 1))) / 2  
 return s  
  
 def range\_(self, b1, b2):  
 list\_ = []  
 i = 2  
 while i <= self.n:  
 ai = self.a + (i - 1) \* self.d  
 list\_ = list\_ + [ai]  
 i += 1  
 if len(list\_) >= b2:  
 return list\_[b1-2:b2]  
 else:  
 while i <= b2:  
 ai = self.a + (i - 1) \* self.d  
 list\_ = list\_ + [ai]  
 i += 1  
 return list\_[b1-2:b2]  
  
 def check\_element(self, x):  
 list\_ = []  
 i = 2  
 while i <= self.n:  
 ai = self.a + (i - 1) \* self.d  
 list\_ = list\_ + [ai]  
 i += 1  
 if x in list\_:  
 return 'True'  
 else:  
 return 'False'  
  
def main():  
 progression = Progression(1,3,10) #create progression where first element=1, difference=3,amount of elements=10  
 print('The seventh element is ',progression.i\_element(7))  
 print('Sum of ten elements equal ',progression.summa(10))  
 print('Range between 3rd and 9th position: ',progression.range\_(3,9))  
 print('Check if it`s "13" in progression:',progression.check\_element(13))  
 print('Check if it`s "14" in progression:', progression.check\_element(14))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Код тестування:

import unittest  
from progression import Progression  
  
  
class TestProgression(unittest.TestCase):  
 test = Progression(1, 3, 10)  
 test2 = Progression(0, 2, 40)  
  
 def test\_i\_element(self):  
 self.assertEqual(self.test.i\_element(8), 22)  
 self.assertEqual(self.test.i\_element(10), 28)  
 self.assertEqual(self.test.i\_element(1), 1)  
 self.assertEqual(self.test2.i\_element(1), 0)  
 self.assertEqual(self.test2.i\_element(10), 18)  
  
 def test\_summa(self):  
 self.assertEqual(self.test.summa(10), 145)  
 self.assertTrue(self.test.summa(3), isinstance(self.test.summa(3), int))  
 self.assertEqual(self.test2.summa(10), 90)  
  
 def test\_range(self):  
 self.assertTrue(self.test.range\_(3, 6), isinstance(self.test.range\_(3, 6), list))  
 self.assertTrue(self.test.range\_(1, 10), isinstance(self.test.range\_(1, 10), list))  
 self.assertTrue(self.test2.range\_(6,9), isinstance(self.test.range\_(1, 10), list))  
  
 def test\_check\_element(self):  
 self.assertEqual(self.test.check\_element(4), 'True')  
 self.assertEqual(self.test.check\_element(5), 'False')  
 self.assertTrue(self.test.check\_element(10), isinstance(self.test.check\_element(10), str))  
 self.assertEqual(self.test2.check\_element(6), 'True')  
 self.assertEqual(self.test2.check\_element(5), 'False')  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

Результат виконання тесту: 

Висновки:

Використання тесту показує, що програма коректно працює з різними числами.