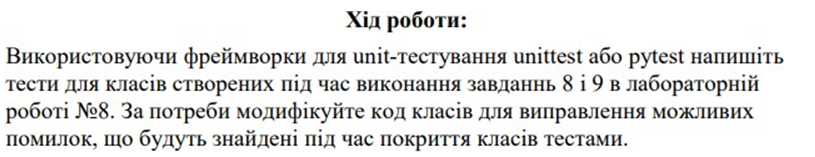
ЛАБОРАТОРНА РОБОТА № 10

unit-тестування в мові Python

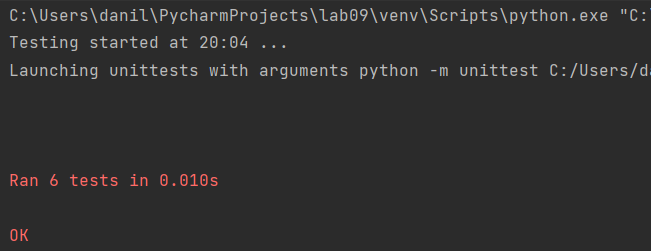
Мета роботи: ознайомитися з фреймворками для unit-тестування в мові Python



Лістинг тесту для 8 завдання:

from task08 import Shop, Discount  
import unittest  
  
  
class TestShop(unittest.TestCase):  
 def setUp(self):  
 self.shop = Shop('Minichek', 'Shop')  
  
 def test\_init(self):  
 self.assertEqual((self.shop.shop\_name, self.shop.shop\_type), ('Minichek', 'Shop'),  
 'Отримані значення є помилковими')  
  
 def test\_set\_number\_of\_units(self):  
 numbers\_of\_units = int('5')  
 self.shop.number\_of\_units = numbers\_of\_units  
 self.assertEqual(self.shop.number\_of\_units, 5)  
  
 def test\_increment\_number\_of\_units(self):  
 numbers\_of\_units = int('5')  
 self.shop.number\_of\_units = numbers\_of\_units  
 increment\_number\_of\_units = int('10')  
 self.shop.number\_of\_units += increment\_number\_of\_units  
 self.assertEqual(self.shop.number\_of\_units, 15)  
  
 def test\_describe\_shop(self):  
 self.assertEqual(f"Назва магазину: {self.shop.shop\_name}\nТип магазину: {self.shop.shop\_type}",  
 self.shop.describe\_shop())  
  
  
class TestDiscount(unittest.TestCase):  
 def setUp(self):  
 list1 = ["Молоко", "Сир", "Ковбаса"]  
 self.discount = Discount(list1, 'Minichek', 'Shop')  
  
 def test\_init(self):  
 self.assertEqual((self.discount.discount\_products, self.discount.shop\_name, self.discount.shop\_type),  
 (["Молоко", "Сир", "Ковбаса"], 'Minichek', 'Shop'),  
 'Отримані значення є помилковими')  
  
 def test\_get\_discounts\_products(self):  
 list = []  
 for i in self.discount.discount\_products:  
 list.append(i)  
 self.assertEqual(f'Знижки на продукти:\n {list}', self.discount.get\_discounts\_products())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

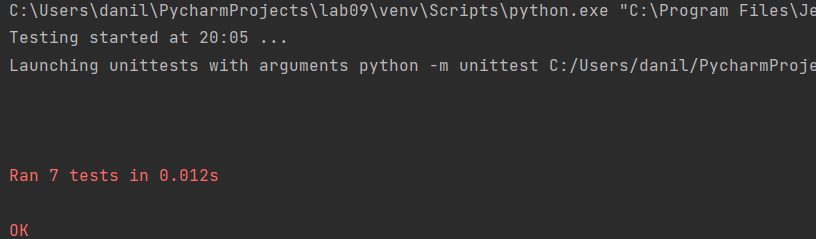
Результат роботи програми:



Лістинг тесту для 9 завдання:

from task09\_adm import Admin  
from task09\_adm import Privileges  
from task09\_user import User  
import unittest  
  
  
class TestUser(unittest.TestCase):  
 def setUp(self):  
 self.user = User("Савченко", "Данило", "Вільський шлях 14", "dan1ssimo\_")  
  
 def test\_init(self):  
 self.assertEqual((self.user.first\_name, self.user.last\_name, self.user.adress, self.user.nickname),  
 ("Савченко", "Данило", "Вільський шлях 14", "dan1ssimo\_"))  
  
 def test\_describe\_user(self):  
 self.assertEqual((self.user.first\_name, self.user.last\_name), self.user.describe\_user())  
  
 def test\_greeting\_user(self):  
 self.assertEqual(f"Вітаю {self.user.nickname}!!!", self.user.greeting\_user())  
  
 def test\_increment\_login\_attempts(self):  
 self.user.login\_attempts += 1  
 self.assertEqual(self.user.increment\_login\_attempts(), self.user.login\_attempts)  
  
 def test\_reset\_login\_attempts(self):  
 self.user.login\_attempts = 0  
 self.assertEqual(self.user.reset\_login\_attempts(), self.user.login\_attempts)  
  
  
class TestAdmin(unittest.TestCase):  
 def setUp(self):  
 list1 = ["Allowed to add message", "Allowed to delete users", "Allowed to ban users"]  
 self.admin = Admin(list1, "Савченко", "Данило", "Вільський шлях 14", "dan1ssimo\_")  
 self.privileges = list1  
  
 def test\_init(self):  
 self.assertEqual(  
 (  
 self.admin.privileges, self.admin.first\_name, self.admin.last\_name, self.admin.adress,  
 self.admin.nickname),  
 (["Allowed to add message", "Allowed to delete users", "Allowed to ban users"], "Савченко", "Данило",  
 "Вільський шлях 14", "dan1ssimo\_"))  
  
 def test\_show\_privileges(self):  
 self.assertEqual(self.admin.privileges, Privileges().show\_privileges())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

Результат роботи програми:



Висновок: ми ознайомилися з фреймворками для unit-тестування в мові Python.