

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №8
дисциплины «Анализ данных»
Вариант 28

Выполнил:
Репкин Александр Павлович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Тестирование в Python [unittest].

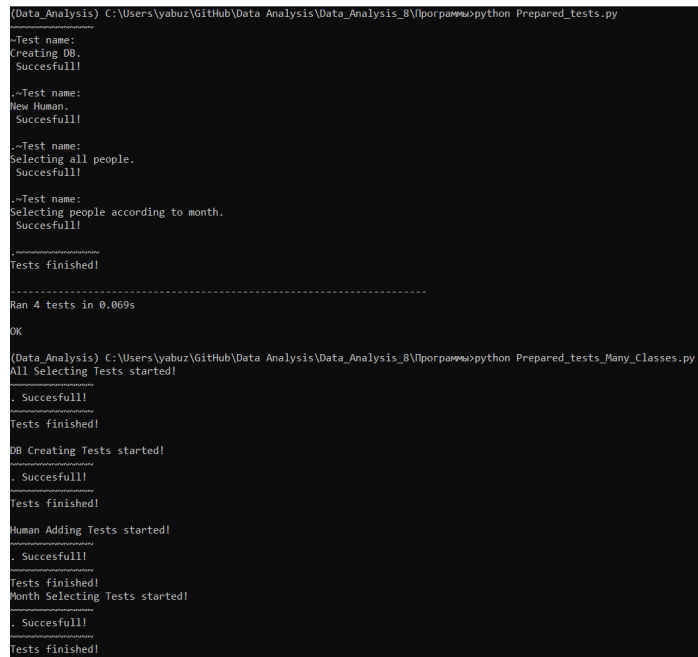
Цель: приобрести навыки написания автоматизированных тестов на языке программирования Python версии 3.x.

Порядок выполнения работы:

1. Выполнено индивидуальное задание – модифицировано индивидуальное задание из работы №7 дисциплины “Анализ Данных”. Были добавлены тесты, позволяющие проверить операции по работе с базой данных SQLite3.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sqlite3
5  from pathlib import Path
6  import Individual as operations
7  import unittest
8
9
10 '''Модернизация индивидуального задания из работы №7.
11 Тестирование операций по работе с базой данных.
12
13 Напоминание: в задании из работы №7: данные о людях хранятся в файле, создаваемом SQLite3 people.db.
14 По умолчанию, файл создаётся в домашнем каталоге пользователя.
15 В данном файле имеется две таблицы: people и surnames'''
16
17
18 class TestDatabaseOperations(unittest.TestCase):
19
20     @classmethod
21     def setUpClass(cls):
22         '''Вызывается один раз для всего класса, в начале.'''
23         print("~~~~~")
24
25     @classmethod
26     def tearDownClass(cls):
27         '''Вызывается один раз для всего класса, в конце.'''
28         print("~~~~~")
29         print("Tests finished!")
30
31     def setUp(self):
32         '''Метод setUp вызывается перед каждым тестом для подготовки окружения.
33 В данном случае он указывает имя для базы данных.'''
34         self.store_tests = Path("test_database.db")
35         print(f"~Test name:")
36
37     def tearDown(self):
38         ...
```

Рисунок 1. Код индивидуального задания



```
(Data_Analysis) C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_8\Программы\python Prepared_tests.py
~~~~~
Test name:
Creating DB.
Successfull!

Test name:
New Human.
Successfull!

Test name:
Selecting all people.
Successfull!

Test name:
Selecting people according to month.
Successfull!

~~~~~
Tests finished!

-----
Ran 4 tests in 0.069s

OK

(Data_Analysis) C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_8\Программы\python Prepared_tests_Many_Classes.py
All Selecting Tests started!
. Successfull!
~~~~~
Tests finished!

DB Creating Tests started!
. Successfull!
~~~~~
Tests finished!

Human Adding Tests started!
. Successfull!
~~~~~
Tests finished!

Month Selecting Tests started!
. Successfull!
~~~~~
Tests finished!
```

Рисунок 2. Пример выполнения программы, 4 успешных теста

2. Ответы на вопросы.

1) Для чего используется автономное тестирование?

Ответ: автономное тестирование используется для автоматизации проверки работоспособности программного обеспечения, позволяя выполнять тесты без необходимости ручного вмешательства.

2) Какие фреймворки Python получили наибольшее распространение для решения задач автономного тестирования?

Ответ: unittest, pytest и nose – наиболее популярные фреймворки Python.

3) Какие существуют основные структурные единицы модуля unittest?

Ответ: основные структурные единицы модуля unittest – тестовые классы, методы тестирования (Начинаются на test_) и assertIn для проверки ожидаемых результатов.

4) Какие существуют способы запуска тестов unittest?

Ответ: тесты unittest можно запускать как из командной строки, так и из среды разработки (Visual Studio Code, PyCharm, Visual Studio и т.д.), а также можно использовать специализированные инструменты для автоматического запуска тестов (Cricket).

5) Каково назначение класса TestCase?

Ответ: класс TestCase определяет тестовые случаи, которые содержат методы для проверки конкретных аспектов работы программы.

6) Какие методы класса TestCase выполняются при запуске и завершении работы тестов?

Ответ: при запуске и завершении работы тестов запускаются методы setUp() и tearDown().

7) Какие методы класса TestCase используются для проверки условий и генерации ошибок?

Ответ: методы assertIn() используются для проверки различных условий и генерации ошибок в случае несоответствия ожидаемых и получаемых результатов.

8) Какие методы класса TestCase позволяют собирать информацию о самом тесте?

Ответ: методы setUp(), tearDown(), getName() и id() позволяют собирать информацию о самом тесте, его имени и идентификаторе.

9) Каково назначение класса TestSuite? Как осуществляется загрузка тестов?

Ответ: класс TestSuite группирует тестовые случаи в единую структуру. Загружаются тесты путём добавления тестовых случаев в объект TestSuite, который выполняется после этого.

10) Каково назначение класса TestResult?

Ответ: класс TestResult отслеживает результаты выполнения тестов. В эту информацию включены сведения о пройденных, проваленных и пропущенных тестах.

11) Для чего может понадобиться пропуск отдельных тестов?

Ответ: пропуск отдельных тестов может понадобиться, когда тестируемый код, временно недоступен или в процессе разработки\доработки.

12) Как выполняется безусловный и условных пропуск тестов? Как выполнить пропуск класса тестов?

Ответ: безусловный пропуск теста выполняется с помощью декоратора @unittest.skip(), а условный пропуск с помощью декоратора @unittest.skipIf(). Для пропуска класса тестов можно использовать декоратор @unittest.skip() перед определением класса.

13) Обобщённый алгоритм проведения тестирования с помощью PyCharm?

Ответ: обобщённый алгоритм проведения тестирования с помощью PyCharm включает: создание тестовых случаев и их методов; настройка окружения тестирования; написание необходимых тестов (И их запуск через PyCharm); анализ результатов и исправление выявленных проблем.

Вывод: в ходе выполнения лабораторной работы, исследовано взаимодействие с базами данных SQLite3 с помощью языка программирования Python.