

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №8
дисциплины «Анализ данных»
Вариант 29

Выполнил:
Саенко Андрей Максимович
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. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____
Ставрополь, 2024 г.

Тема: Тестирование в Python [unittest]

Цель: приобретение навыков написания автоматизированных тестов на языке программирования Python версии 3.x.

Порядок выполнения работы:

Задание 1. Для индивидуального задания лабораторной работы 2.21 добавьте тесты с использованием модуля unittest, проверяющие операции по работе с базой данных.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sqlite3
from pathlib import Path
import ind
import unittest

# Для индивидуального задания лабораторной
# работы 2.21 добавьте тесты с использованием
# модуля unittest, проверяющие операции по работе
# с базой данных.

class CreateDBtests(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        print("\nТестирование функции, создающей базу данных.")
        print("-----")

    @classmethod
    def tearDownClass(cls):
        print("OK")
        print("-----")

    def setUp(self):
        self.store_tests = Path("create_test.db")

    def tearDown(self):
        if self.store_tests.exists():
            conn = sqlite3.connect(self.store_tests)
            conn.close()
            self.store_tests.unlink()

    def test_create_db(self):
        """Тестирование работы функции, создающей базу данных."""
        ind.create_db(self.store_tests)
```

```

conn = sqlite3.connect(self.store_tests)
cursor = conn.cursor()
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
tables = cursor.fetchall()
self.assertIn('routes', tables)
self.assertIn('end_stations', tables)
conn.close()

```

```

class AddTest(unittest.TestCase):

```

```

    @classmethod
    def setUpClass(cls):
        print("\nТестирование функции добавления маршрута.")
        print("-----")

```

```

    @classmethod
    def tearDownClass(cls):
        print("OK")
        print("-----")

```

```

    def setUp(self):
        self.db_path = Path("test_add.db")

```

```

    def tearDown(self):
        if self.db_path.exists():
            conn = sqlite3.connect(self.db_path)
            conn.close()
            self.db_path.unlink()

```

```

    def test_add_route(self):
        """Тестирование работы функции, добавляющей маршрут в базу данных."""
        ind.create_db(self.db_path)
        ind.add_route(self.db_path, "Stavropol",
                      "Krasnodar", 1)

```

```

        conn = sqlite3.connect(self.db_path)
        cursor = conn.cursor()
        cursor.execute(
            """
            SELECT routes.start_name, end_stations.station_title,
            routes.route_number
            FROM routes
            INNER JOIN end_stations ON
            end_stations.station_id = routes.station_id
            """
        )
        route = cursor.fetchone()
        self.assertIsNotNone(route)
        self.assertEqual(route[0], "Stavropol")
        self.assertEqual(route[1], "Krasnodar")
        self.assertEqual(route[2], 1)

```

```

conn.close()

class SelectTest(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        print("\nТестирование функций выделения маршрутов.")
        print("-----")

    @classmethod
    def tearDownClass(cls):
        print("OK")
        print("-----")

    def setUp(self):
        self.db_path = Path("select_tests.db")
        ind.create_db(self.db_path)

    def tearDown(self):
        if self.db_path.exists():
            conn = sqlite3.connect(self.db_path)
            conn.close()
            self.db_path.unlink()

    def test_select_routes(self):
        """Тестирование вывода маршрутов с заданным параметром."""
        ind.add_route(self.db_path, "A", "B", 1)
        ind.add_route(self.db_path, "B", "C", 2)
        ind.add_route(self.db_path, "C", "D", 3)
        routes = ind.select_routes(self.db_path, "B")
        self.assertEqual(len(routes), 2)
        self.assertEqual(routes[0]["start"], "A")
        self.assertEqual(routes[0]["end"], "B")
        self.assertEqual(routes[0]["number"], 1)
        self.assertEqual(routes[1]["start"], "B")
        self.assertEqual(routes[1]["end"], "C")
        self.assertEqual(routes[1]["number"], 2)

    def test_select_all(self):
        """Тестирование вывода всех маршрутов."""
        ind.add_route(self.db_path, "A", "B", 1)
        ind.add_route(self.db_path, "C", "D", 2)
        routes = ind.select_all(self.db_path)
        self.assertEqual(len(routes), 2)
        self.assertEqual(routes[0]["start"], "A")
        self.assertEqual(routes[0]["end"], "B")
        self.assertEqual(routes[0]["number"], 1)
        self.assertEqual(routes[1]["start"], "C")
        self.assertEqual(routes[1]["end"], "D")
        self.assertEqual(routes[1]["number"], 2)

if __name__ == '__main__':

```

unittest.main()

Результат работы программы:

```
(Data_Analysis) C:\Users\HAIER\Desktop\Задания\Анализ данных\Data_lr_8\code>python -m unittest -v ind_tests.py
Тестирование функции добавления маршрута.
-----
test_add_route (ind_tests.AddTest.test_add_route)
Тестирование работы функции, добавляющей маршрут в базу данных. ... ok
OK
-----
Тестирование функции, создающей базу данных.
-----
test_create_db (ind_tests.CreateDBtests.test_create_db)
Тестирование работы функции, создающей базу данных. ... ok
OK
-----
Тестирование функций выделения маршрутов.
-----
test_select_all (ind_tests.SelectTest.test_select_all)
Тестирование вывода всех маршрутов. ... ok
test_select_routes (ind_tests.SelectTest.test_select_routes)
Тестирование вывода маршрутов с заданным параметром. ... ok
OK
-----
Ran 4 tests in 0.071s
```

Рисунок 1 – Тестирование функций программы

Выводы: в ходе выполнения работы получены навыки тестирования функций с помощью модуля unittest.