

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»

Выполнил:
Середа Кирилл Витальевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(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.

Ход работы:

Индивидуальное задание

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

Создал файл ind_test.py, в котором написал различные тесты для проверки работы 2.21: проверка на создание бд, проверка на добавление элемента в бд, проверка на валидность количества столбцов, на рис. 2 представлен код/

Также создал файл ind_runner.py, который запускает тест из файла ind_test.py, на рис. 1 представлен код

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

import unittest
import ind_tests

if __name__ == "__main__":
    ind_test_suite = unittest.TestSuite()
    ind_test_suite.addTest(unittest.makeSuite(ind_tests.indTest))
    runner = unittest.TextTestRunner(verbosity=2)
    runner.run(ind_test_suite)
```

Рисунок 1 – Код ind_runner.py

```

class indTest(unittest.TestCase):
    @classmethod
    def setUpClass(cls):
        """Set up for class"""
        print("Проверка работы операций с базами данных")
        print("=====")

    @classmethod
    def tearDownClass(cls):
        """Tear down for class"""
        print("=====")
        print("Конец")

    def test_create_bd(self):
        def created_bd(name_bd):
            file_path = pathlib.Path.cwd() / name_bd

            if file_path.exists and file_path.is_file:
                return True
            return False

        self.assertEqual(ind.create_db("test_bd"), created_bd("test_bd"))

    def test_add_person(self):
        def added_person(name_bd, full_name, birth_date, phone):
            conn = sqlite3.connect(name_bd)
            cursor = conn.cursor()

            cursor.execute(
                """
                SELECT names.full_name, person.birth_date, person.phone_number
                FROM person
                INNER JOIN names ON names.name_id = person.name_id
                WHERE
                names.full_name = ? and
                person.birth_date = ? and
                person.phone_number = ?
                """,
                (full_name, birth_date, phone),
            )
            rows = cursor.fetchall()
            conn.close()

            return bool(rows)

        ind.add_person("test_bd", "Артём", 22, 5555)
        self.assertEqual(
            True,
            added_person("test_bd", "Артём", 22, 5555),
        )

    def test_select_all_last(self):

```

Рисунок 2 – Результат выполнения тестов

Вывод: в ходе выполнения работы были приобретены навыки написания автоматизированных тестов на языке программирования Python версии 3.x.