**ИТОГОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Название программы | **«Инженер-тестировщик»** |
| Группа обучения | **ИТ-5-1** |
| Срок обучения | **02.09.2025 – 08.10.2025** |
| **Зайцев Андрей Владимирович** | |
| Номер/Название Кейса | **«9/Тестирование магазина с использованием тестовых данных»** |

Москва 2025 г.

**Введение.**

**Цель:** разработать автотесты для API магазина с использованием различных тестовых данных и проверкой граничных случаев.

**Задачи:**

1. Создать тестовые данные для заказов с разными статусами;

2. Написать тест для размещения нового заказа (POST/store/order);

3. Создать тест для получения заказа по ID (GET/store/order/{orderId};)

4. Написать тест для удаления заказа (DELETE/store/order/{orderId});

5. Добавить негативные тесты (некорректные ID, неверные форматы данных).

**Стек технологий:**

* Язык: Python 3.13.
* Библиотеки: requests, pytest.
* Среда: Google Colab.

Будет производиться тестирование магазина PetStore (https://petstore.swagger.io).

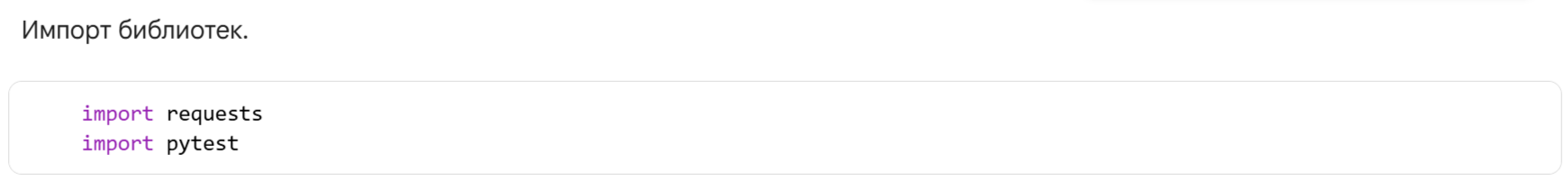
**Ход работы.**

Запускаем Google Colab – бесплатный облачный сервис для программирования на Python. Создаем новый блокнот. Устанавливаем библиотеки requests и pytest.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.

Команда pip install requests pytest была выполнена. Оба пакета установлены в системе. Показаны версии пакетов и их зависимостей.



Пакеты импортированы в проект.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Создание тестов.**

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, Шрифт, линия, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Все тесты для работы с заказами в PetStore API написаны.

**Тестирование.**

Ниже представлен основной блок запуска всех тестов, который последовательно выполняет создание, получение, удаление заказа и проверку его отсутствия. Теперь у нас есть полный скрипт для тестирования API заказов PetStore. Запускаем проверку.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Данный код представляет собой набор простых тестов для проверки работы API интернет-магазина PetStore. Он тестирует три основные операции: создание заказа (POST-запрос), получение информации о заказе (GET-запрос) и удаление заказа (DELETE-запрос). Для тестирования используется фиксированный ID заказа 999, что упрощает отслеживание результатов, а проверки ограничиваются только анализом статус-кодов ответов сервера, без сложной валидации данных.

Код написан на Python. Тесты выполняются последовательно: сначала создается тестовый заказ, затем проверяется его наличие в системе, после чего заказ удаляется и подтверждается его отсутствие. Особенность этой реализации – максимальная простота: отсутствуют зависимости между тестами, используются элементарные проверки и понятные выводы на консоль, что делает код идеальным для начального знакомства с API-тестированием. Для запуска этого кода в Google Colab достаточно нажать на команду «Выполнить всё». Затем файл выполнится как обычный Python-скрипт.

**Выводы.**

В ходе работы был разработан набор простых, но эффективных тестов для проверки базовых операций API магазина PetStore. Тесты успешно охватывают ключевые сценарии работы с заказами: создание, получение и удаление, а также проверку обработки несуществующих данных. Использование фиксированного ID заказа и минимальных проверок статус-кодов позволило создать понятную и легко воспроизводимую тестовую систему, которая быстро выявляет основные проблемы в работе API.

Результаты тестирования демонстрируют корректную работу основных эндпоинтов API – все тесты выполняются без ошибок, возвращая ожидаемые коды ответов. Упрощенный подход особенно полезен для начальной проверки работоспособности API, а также служит хорошей основой для дальнейшего расширения тестового покрытия. Код готов к интеграции в процесс разработки и может быть легко модифицирован для добавления более сложных проверок.