



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Информационная безопасность»

ОТЧЕТ
по лабораторной работе № 3
по курсу «Искусственный интеллект»
на тему: «Тестирование с Pytest»
Вариант № 6

Студент ИУ8-13М
(Группа)

(Подпись, дата)

Савватеев А. Э.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Зотов М. В.
(И. О. Фамилия)

2025 г.

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ	3
2 ТРЕБОВАНИЯ ЛАБОРАТОРНОЙ РАБОТЫ	4
3 ХОД РАБОТЫ	5
3.1 Тестируемые модели	5
3.2 Структура тестов	5
3.3 Особенности реализации тестов	6
3.4 Пример структуры теста	6
3.5 Тестовые данные (YAML)	6
3.6 GitHub Actions	7
4 ЗАКЛЮЧЕНИЕ	8

1 ЦЕЛЬ РАБОТЫ

Создать комплексную систему тестирования для Pydantic моделей с использованием pytest, YAML-файлов с тестовыми данными и автоматизации через tox и GitHub Actions.

2 ТРЕБОВАНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

- Созданы тесты для всех Pydantic моделей
- Тесты используют фикстуры и параметризацию
- Тестовые данные вынесены в YAML-файлы
- Настроен pytest с маркерами для группировки тестов
- Добавлен GitHub Action для автоматического запуска тестов
- Настроен tox для изолированного окружения тестирования

3 ХОД РАБОТЫ

3.1 Тестируемые модели

В лабораторной работе №2 были реализованы следующие Pydantic модели в файле `students_folder/savandr/lab_2/lab_2.py`:

- **UserSpec** — модель профиля пользователя с валидацией email и проверкой русских символов
- **ProfileSpec** — расширенная модель пользователя с полями bio и url
- **ItemSpec** — модель товара с валидацией цены
- **ServiceSpec** — модель услуги с валидацией
- **OrderLineSpec** — модель строки заказа с вычисляемым полем
- **OrderSpec** — модель заказа
- **OrdersSpec** — модель списка заказов

3.2 Структура тестов

Реализована полная система тестирования в директории `tests/Savvateev/lab_3/`:

- `test_user_spec.py` — тесты для модели `UserSpec`
- `test_profile_spec.py` — тесты для модели `ProfileSpec`
- `test_item_spec.py` — тесты для модели `ItemSpec`
- `test_service_spec.py` — тесты для модели `ServiceSpec`
- `test_order_line_spec.py` — тесты для модели `OrderLineSpec`
- `test_order_spec.py` — тесты для модели `OrderSpec`
- `test_orders_spec.py` — тесты для модели `OrdersSpec`

3.3 Особенности реализации тестов

- Все тесты помечены маркером `@pytest.mark.sae_lab3`
- Используется фикстура `yaml_test_data` из `conftest.py`
- Тестовые данные хранятся в YAML-файлах в `tests/src/Savvateev/lab_3/`
- Каждый тест проверяет как валидные, так и невалидные данные
- Используется параметризация через YAML для множественных тест-кейсов

3.4 Пример структуры теста

```
@pytest.mark.sae_lab3
def test_class_user_spec(yaml_test_data):
    test_cases = yaml_test_data("Savvateev/lab_3/UserSpec.yaml")
    for test_desc, value, answer in test_cases:
        try:
            test_class = UserSpec.model_validate(value)
            test_answer = True
            assert answer == test_answer
        except ValidationError as e:
            test_answer = False
            assert answer == test_answer
```

3.5 Тестовые данные (YAML)

В директории `tests/src/Savvateev/lab_3/` созданы YAML-файлы с тестовыми данными:

- `UserSpec.yaml` — данные для тестирования пользователей
- `ProfileSpec.yaml` — данные для тестирования профилей
- `ItemSpec.yaml` — данные для тестирования товаров
- `ServiceSpec.yaml` — данные для тестирования услуг
- `OrderLineSpec.yaml` — данные для тестирования строк заказа

- `OrderSpec.yaml` — данные для тестирования заказов
- `OrdersSpec.yaml` — данные для тестирования списков заказов

Каждый YAML-файл содержит набор тест-кейсов.

3.6 GitHub Actions

Создан workflow-файл `.github/workflows/sae-lab3.yml`, который:

- Запускается при pull request с изменениями в путях:
 - `students_folder/savandr/**`
 - `tests/Savvateev/**`
 - `tests/src/Savvateev/**`
- Использует Python 3.13
- Устанавливает `tox`
- Запускает тесты через `tox -e sae_lab3`

4 ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была реализована комплексная система тестирования с использованием pytest, YAML-данных, автоматизации через tox и GitHub Actions. Все основные требования выполнены. Предложенные рекомендации направлены на дальнейшее улучшение качества тестирования.