

# Практическая работа №4

## Тестирование расписания

ЗАДАНИЕ: получаем ваше расписание на сайте <https://api.ntnu.ru/> и сверяем тестами pytest с заранее выбранным. Показываем что если заменяем свое расписание то тесты не проходят.

## Результаты тестирования

```
C:\Users\danil\Desktop\Управленческая>pytest test_ntnu_schedule.py -v
===== test session starts =====
platform win32 -- Python 3.13.7, pytest-7.4.0, pluggy-1.6.0 -- C:\Users\danil\AppData\Local\Programs\Python\Python313\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\danil\Desktop\Управленческая
collected 17 items

test_ntnu_schedule.py::TestNTNUSchedule::test_api_availability PASSED [ 59%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_api_endpoint PASSED [ 71%]
test_ntnu_schedule.py::TestNTNUSchedule::test_my_schedule_structure PASSED [ 77%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_content_integrity PASSED [ 82%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_modification_detection PASSED [ 88%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_time_format PASSED [ 94%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_teachers_exist PASSED [ 99%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_rooms_exist PASSED [ 100%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_subjects_exist PASSED [ 100%]
test_ntnu_schedule.py::TestNTNUSchedule::test_web_interface_loading PASSED [ 100%]
DevTools Listening on ws://127.0.0.1:62886/devtools/browser/832ef386-6754-416d-8950-bb76d58abae3 [ 50%]
test_ntnu_schedule.py::TestNTNUSchedule::test_web_schedule_presence PASSED [ 60%]
DevTools Listening on ws://127.0.0.1:62887/devtools/browser/31d6d422-9376-4b3c-9691-475e9b67952b [ 60%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_consistency PASSED [ 70%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_completeness PASSED [ 76%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_realistic_times PASSED [ 82%]
test_ntnu_schedule.py::TestNTNUSchedule::test_compare_with_api_schedule SKIPPED (API недоступна для сравнения) [ 88%]
test_ntnu_schedule.py::test_single_http_check PASSED [ 94%]
test_ntnu_schedule.py::test_schedule_demo PASSED [ 100%]

===== 16 passed, 1 skipped in 25.82s =====
```

Рисунок 1- Результаты успешного запуска тестирования

```
C:\Users\danil\Desktop\Управленческая>pytest test_ntnu_schedule.py -v
===== test session starts =====
platform win32 -- Python 3.13.7, pytest-7.4.0, pluggy-1.6.0 -- C:\Users\danil\AppData\Local\Programs\Python\Python313\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\danil\Desktop\Управленческая
collected 17 items

test_ntnu_schedule.py::TestNTNUSchedule::test_api_availability PASSED [ 59%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_api_endpoint PASSED [ 71%]
test_ntnu_schedule.py::TestNTNUSchedule::test_my_schedule_structure PASSED [ 77%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_content_integrity PASSED [ 82%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_modification_detection FAILED [ 88%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_time_format PASSED [ 94%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_teachers_exist PASSED [ 99%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_rooms_exist PASSED [ 100%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_subjects_exist PASSED [ 100%]
test_ntnu_schedule.py::TestNTNUSchedule::test_web_interface_loading PASSED [ 100%]
DevTools Listening on ws://127.0.0.1:54093/devtools/browser/493c0aea-207e-4a41-ab77-9a3d5591b1f9 [ 50%]
test_ntnu_schedule.py::TestNTNUSchedule::test_web_schedule_presence PASSED [ 60%]
DevTools Listening on ws://127.0.0.1:54092/devtools/browser/c75c79e-78f9-4b18-948c-2b8dc4d43669 [ 60%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_consistency PASSED [ 70%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_completeness PASSED [ 76%]
test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_realistic_times PASSED [ 82%]
test_ntnu_schedule.py::TestNTNUSchedule::test_compare_with_api_schedule SKIPPED (API недоступна для сравнения) [ 88%]
test_ntnu_schedule.py::test_single_http_check PASSED [ 94%]
test_ntnu_schedule.py::test_schedule_demo PASSED [ 100%]

===== 16 passed, 1 skipped, 1 failure in 25.82s =====
TestNTNUSchedule.test_schedule_modification_detection

self = <test_ntnu_schedule.TestNTNUSchedule object at 0x0000015015D16780>
my_schedule = {'Friday': [{'room': '220', 'subject': 'Корпоративные информационные системы', 'teacher': 'Гуськова Елена Александровна...'}], 'Monday': [{'room': '220', 'subject': 'Корпоративные информационные системы', 'teacher': 'Комаров Александр Олегович', 'time': '15:30-17:00'}], ...}]

def test_schedule_modification_detection(self, my_schedule):
    original_subject = my_schedule['Monday'][0]['subject']

    modified_schedule = my_schedule.copy()
    modified_schedule['Monday'] = my_schedule['Monday'].copy()
    modified_schedule['Monday'][0] = my_schedule['Monday'][0].copy()
    modified_schedule['Monday'][0]['subject'] = 'Измененный предмет'

    assert modified_schedule['Monday'][0]['subject'] != original_subject
    assert modified_schedule['Monday'][0]['subject'] == 'Измененный предмет'
    assert original_subject == 'Корпоративные информационные системы'
    AssertionError: assert 'ИЗМЕНЕННЫЙ ПРЕДМЕТ' == 'Корпоративные...онные системы'
    + корпоративные информационные системы
    - ИЗМЕНЕННЫЙ ПРЕДМЕТ

test_ntnu_schedule.py:118: AssertionError
===== short test summary info =====
FAILED test_ntnu_schedule.py::TestNTNUSchedule::test_schedule_modification_detection - AssertionError: assert 'ИЗМЕНЕННЫЙ ПРЕДМЕТ' == 'Корпоративные...онные системы'
1 failed, 16 passed, 1 skipped in 25.82s
```

Рисунок 2- Неправильный результат(изменил значения в расписании)

```
@pytest.fixture
def my_schedule(self):
    return {
        "Monday": [
            {"time": "08:30-10:00", "subject": "ИЗМЕНЕННЫЙ ПРЕДМЕТ", "teacher": "Неизвестный Преподаватель", "room": "999"},
            {"time": "10:10-11:40", "subject": "Корпоративные информационные системы", "teacher": "Гуськова Юлия Александровна", "room": "218"},
            {"time": "12:10-13:40", "subject": "Корпоративные информационные системы", "teacher": "Гуськова Юлия Александровна", "room": "218"},
            {"time": "13:50-15:20", "subject": "Организационно-экономическое обоснование системы и технических решений", "teacher": "Гусева Ирина Борисовна", "room": "220"},
            {"time": "15:30-17:00", "subject": "Организационно-экономическое обоснование системы и технических решений", "teacher": "Гусева Ирина Борисовна", "room": "220"}
        ],
    }
```

Рисунок 3- Измененные значения

Код файла с тестами (test\_nntu\_schedule.py):

```
import pytest
import requests
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

class TestNGTUSchedule:

    @pytest.fixture
    def driver(self):
        driver = webdriver.Chrome()
        driver.implicitly_wait(10)
        yield driver
        driver.quit()

    @pytest.fixture
    def my_schedule(self):
        return {
            "monday": [
                {"time": "08:30-10:00", "subject": "ИЗМЕНЕННЫЙ ПРЕДМЕТ",
"teacher": "Неизвестный Преподаватель", "room": "999"},
                {"time": "10:10-11:40", "subject": "Корпоративные информационные
системы", "teacher": "Гуськова Юлия Александровна", "room": "218"},
                {"time": "12:10-13:40", "subject": "Корпоративные информационные
системы", "teacher": "Гуськова Юлия Александровна", "room": "218"},
                {"time": "13:50-15:20", "subject": "Организационно-экономическое
обоснование системы и технических решений", "teacher": "Гусева Ирина Борисовна",
"room": "220"},
                {"time": "15:30-17:00", "subject": "Организационно-экономическое
обоснование системы и технических решений", "teacher": "Гусева Ирина Борисовна",
"room": "220"}
            ],
            "tuesday": [
                {"time": "08:30-10:00", "subject": "Надежность и
отказоустойчивость информационных систем", "teacher": "Жидкова Наталья
Валерьевна", "room": "218"},
                {"time": "10:10-11:40", "subject": "Надежность и
отказоустойчивость информационных систем", "teacher": "Жидкова Наталья
Валерьевна", "room": "218"},
                {"time": "12:10-13:40", "subject": "Надежность и
отказоустойчивость информационные системы", "teacher": "Жидкова Наталья
Валерьевна", "room": "218"}
            ],
            "wednesday": [
                {"time": "10:10-11:40", "subject": "Корпоративные информационные
системы", "teacher": "Гуськова Юлия Александровна", "room": "220"},
                {"time": "12:10-13:40", "subject": "Корпоративные информационные
системы", "teacher": "Гуськова Юлия Александровна", "room": "220"},
            ]
        }
```

```

        {"time": "13:50-15:20", "subject": "Корпоративные информационные системы", "teacher": "Гуськова Юлия Александровна", "room": "220"}
    ],
    "thursday": [
        {"time": "08:30-10:00", "subject": "Эксплуатация и модификация информационных систем", "teacher": "Жидкова Наталья Валерьевна", "room": "210"},
        {"time": "10:10-11:40", "subject": "Эксплуатация и модификация информационных систем", "teacher": "Жидкова Наталья Валерьевна", "room": "210"},
        {"time": "12:10-13:40", "subject": "Эксплуатация и модификация информационных систем", "teacher": "Жидкова Наталья Валерьевна", "room": "210"},
        {"time": "13:50-15:20", "subject": "Основы тестирования программного обеспечения", "teacher": "Комаров Александр Олегович", "room": "320"},
        {"time": "15:30-17:00", "subject": "Основы тестирования программного обеспечения", "teacher": "Комаров Александр Олегович", "room": "320"}
    ],
    "friday": [
        {"time": "08:30-10:00", "subject": "Корпоративные информационные системы", "teacher": "Гуськова Юлия Александровна", "room": "220"},
        {"time": "10:10-11:40", "subject": "Корпоративные информационные системы", "teacher": "Гуськова Юлия Александровна", "room": "220"},
        {"time": "12:10-13:40", "subject": "Корпоративные информационные системы", "teacher": "Гуськова Юлия Александровна", "room": "220"},
        {"time": "13:50-15:20", "subject": "Основы тестирования программного обеспечения", "teacher": "Комаров Александр Олегович", "room": "320"}
    ],
    "saturday": [
        {"time": "08:30-10:00", "subject": "Теория цифровой обработки сигналов", "teacher": "Абаимов Анатолий Вячеславович", "room": "320"},
        {"time": "10:10-11:40", "subject": "Теория цифровой обработки сигналов", "teacher": "Абаимов Анатолий Вячеславович", "room": "320"},
        {"time": "12:10-13:40", "subject": "Теория цифровой обработки сигналов", "teacher": "Абаимов Анатолий Вячеславович", "room": "320"},
        {"time": "13:50-15:20", "subject": "Теория цифровой обработки сигналов", "teacher": "Абаимов Анатолий Вячеславович", "room": "320"},
        {"time": "15:30-17:00", "subject": "Теория цифровой обработки сигналов", "teacher": "Абаимов Анатолий Вячеславович", "room": "317"}
    ]
}

```

```

def test_api_availability(self):
    response = requests.get('https://api.nntu.ru/')
    assert response.status_code == 200

```

```

def test_schedule_api_endpoint(self):
    endpoints = [
        'https://api.nntu.ru/content/schedule',
        'https://api.nntu.ru/schedule',
        'https://api.nntu.ru/api/schedule'
    ]

```

```

]

for endpoint in endpoints:
    try:
        response = requests.get(endpoint, timeout=5)
        if response.status_code == 200:
            assert response.status_code == 200
            return
    except:
        continue

pytest.skip("API endpoint для расписания не найден")

def test_my_schedule_structure(self, my_schedule):
    assert isinstance(my_schedule, dict)
    days = ['monday', 'tuesday', 'wednesday', 'thursday', 'friday',
'saturday']

    for day in days:
        assert day in my_schedule
        assert isinstance(my_schedule[day], list)

        for class_item in my_schedule[day]:
            assert 'time' in class_item
            assert 'subject' in class_item
            assert 'teacher' in class_item
            assert 'room' in class_item

def test_schedule_content_integrity(self, my_schedule):
    for day, classes in my_schedule.items():
        for class_item in classes:
            assert class_item['time'].count('-') == 1
            assert ':' in class_item['time']
            assert len(class_item['subject']) > 5
            assert len(class_item['teacher']) > 5
            assert len(class_item['room']) >= 2

def test_schedule_modification_detection(self, my_schedule):
    original_subject = my_schedule['monday'][0]['subject']

    modified_schedule = my_schedule.copy()
    modified_schedule['monday'] = my_schedule['monday'].copy()
    modified_schedule['monday'][0] = my_schedule['monday'][0].copy()
    modified_schedule['monday'][0]['subject'] = 'Измененный предмет'

    assert modified_schedule['monday'][0]['subject'] != original_subject
    assert modified_schedule['monday'][0]['subject'] == 'Измененный предмет'
    assert original_subject == 'Корпоративные информационные системы'

def test_schedule_time_format(self, my_schedule):
    for day, classes in my_schedule.items():

```

```

        for class_item in classes:
            time_str = class_item['time']
            assert isinstance(time_str, str)
            assert len(time_str.split('-')) == 2
            assert ':' in time_str

def test_schedule_teachers_exist(self, my_schedule):
    for day, classes in my_schedule.items():
        for class_item in classes:
            assert 'teacher' in class_item
            assert isinstance(class_item['teacher'], str)
            assert len(class_item['teacher'].strip()) > 0

def test_schedule_rooms_exist(self, my_schedule):
    for day, classes in my_schedule.items():
        for class_item in classes:
            assert 'room' in class_item
            assert isinstance(class_item['room'], str)
            assert len(class_item['room'].strip()) > 0

def test_schedule_subjects_exist(self, my_schedule):
    for day, classes in my_schedule.items():
        for class_item in classes:
            assert 'subject' in class_item
            assert isinstance(class_item['subject'], str)
            assert len(class_item['subject'].strip()) > 0

def test_web_interface_loading(self, driver):
    driver.get('https://www.nntu.ru/')

    wait = WebDriverWait(driver, 15)
    body = wait.until(EC.presence_of_element_located((By.TAG_NAME, 'body')))

    assert driver.title != ''
    assert 'ннгу' in driver.title.lower()

def test_web_schedule_presence(self, driver):
    driver.get('https://www.nntu.ru/')

    schedule_keywords = ['расписание', 'schedule', 'пара', 'занятия',
                          'lecture']
    page_text = driver.page_source.lower()

    has_schedule_content = any(keyword in page_text for keyword in
                                schedule_keywords)
    assert has_schedule_content == True

def test_schedule_consistency(self, my_schedule):
    schedule_copy = my_schedule.copy()
    assert schedule_copy == my_schedule

```

```

def test_schedule_completeness(self, my_schedule):
    total_classes = sum(len(classes) for classes in my_schedule.values())
    assert total_classes > 0
    assert total_classes >= 20

def test_schedule_realistic_times(self, my_schedule):
    for day, classes in my_schedule.items():
        for class_item in classes:
            time_parts = class_item['time'].split('-')
            assert len(time_parts) == 2
            start_time, end_time = time_parts

            assert len(start_time.split(':')) == 2
            assert len(end_time.split(':')) == 2

            start_minutes = int(start_time.split(':')[0]) * 60 +
int(start_time.split(':')[1])
            end_minutes = int(end_time.split(':')[0]) * 60 +
int(end_time.split(':')[1])
            assert start_minutes < end_minutes

def test_compare_with_api_schedule(self, my_schedule):
    try:
        endpoints = [
            'https://api.nntu.ru/content/schedule',
            'https://api.nntu.ru/schedule',
            'https://api.nntu.ru/api/schedule'
        ]

        for endpoint in endpoints:
            try:
                response = requests.get(endpoint, timeout=5)
                if response.status_code == 200:
                    api_schedule = response.json()
                    assert isinstance(api_schedule, (dict, list))
                    return
            except:
                continue

        pytest.skip("API недоступно для сравнения")

    except requests.exceptions.RequestException:
        pytest.skip("API недоступно для сравнения")

def test_simple_http_check():
    response = requests.get('https://www.nntu.ru/')
    assert response.status_code == 200
    print("Сайт НГТУ доступен!")

def test_schedule_demo():
    schedule = {

```

```
        "monday": [  
            {"time": "08:30-10:00", "subject": "Корпоративные информационные  
системы", "teacher": "Гуськова Ю.А.", "room": "218"},  
            {"time": "10:10-11:40", "subject": "Корпоративные информационные  
системы", "teacher": "Гуськова Ю.А.", "room": "218"}  
        ]  
    }  
  
    assert len(schedule["monday"]) == 2  
    assert schedule["monday"][0]["subject"] == "Корпоративные информационные  
системы"  
    print("Демонстрационное расписание корректно!")  
  
if __name__ == "__main__":  
    test_simple_http_check()  
    test_schedule_demo()
```