

МИНОБР НАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
им. Р.Е. Алексеева»
АРЗАМАССКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
(ФИЛИАЛ)

Отчёт по лабораторной работе №4

по предмету:

«Тестирование ПО»

Выполнили студенты группы АЗИС 22-2

Кривоногова Е. А.

Паничкина В.Н.

Проверил: Комаров А.О.

Арзамас 2025

```

main x
1 import pytest
2 import requests
3 from selenium import webdriver
4 from selenium.webdriver.common.by import By
5 from selenium.webdriver.support.ui import WebDriverWait
6 from selenium.webdriver.support import expected_conditions as EC
7 from selenium.webdriver.chrome.options import Options
8 import json
9 from datetime import datetime, timedelta
10 import time
11
12
13 class ScheduleTester:
14     def __init__(self):
15         self.base_url = "https://api.ntu.ru"
16         self.expected_schedule = None
17
18     def load_expected_schedule(self, schedule_data):
19         """Загружает ожидаемое расписание"""
20         self.expected_schedule = schedule_data
21
22     def get_current_schedule_from_api(self, group_id=None):
23         """Получает текущее расписание через API"""
24         try:
25             if group_id:
26                 url = f"{self.base_url}/backend/v1/schedule/groups/{group_id}"
27             else:
28                 # Или используем другой endpoint в зависимости от API
29                 url = f"{self.base_url}/backend/v1/schedule"
30
31             response = requests.get(url, timeout=10)
32             response.raise_for_status()
33             return response.json()
34         except requests.RequestException as e:
35             print(f"Ошибка при получении расписания через API: {e}")
36             return None
37
38     def extract_schedule_from_website(self, driver, group_selector):
39         """Извлекает расписание с веб-сайта с помощью Selenium"""

```

Error: The following package(s) are not supported:
selenium

```

main x
41 # Ожидаем загрузки расписания
42 schedule_element = WebDriverWait(driver, 10).until(
43     EC.presence_of_element_located((By.CLASS_NAME, group_selector))
44 )
45
46 # Здесь нужно адаптировать под конкретную структуру сайта
47 schedule_data = {}
48
49 # Пример извлечения данных (нужно адаптировать под реальный сайт)
50 days = driver.find_elements(By.CLASS_NAME, "schedule-day")
51
52 for day in days:
53     day_name = day.find_element(By.CLASS_NAME, "day-name").text
54     lessons = []
55
56     lesson_elements = day.find_elements(By.CLASS_NAME, "lesson")
57     for lesson in lesson_elements:
58         lesson_info = {
59             'time': lesson.find_element(By.CLASS_NAME, "time").text,
60             'subject': lesson.find_element(By.CLASS_NAME, "subject").text,
61             'teacher': lesson.find_element(By.CLASS_NAME, "teacher").text,
62             'room': lesson.find_element(By.CLASS_NAME, "room").text
63         }
64         lessons.append(lesson_info)
65
66     schedule_data[day_name] = lessons
67
68     return schedule_data
69
70 except Exception as e:
71     print(f"Ошибка при извлечении расписания: {e}")
72     return None
73
74
75 class TestSchedule:
76
77     @pytest.fixture
78     def chrome_driver(self):
79         """Фикстура для создания Chrome вкладки"""

```

Error: The following package(s) are not supported:
selenium

```

main x
78 def chrome_driver(self):
79     """Фикстура для создания Chrome драйвера"""
80     chrome_options = Options()
81     chrome_options.add_argument("--headless") # Раскомментировать для CI/CD
82     chrome_options.add_argument("--no-sandbox")
83     chrome_options.add_argument("--disable-dev-shm-usage")
84
85     driver = webdriver.Chrome(options=chrome_options)
86     yield driver
87     driver.quit()
88
89 @pytest.fixture
90 def schedule_tester(self):
91     """Фикстура для создания тестера расписания"""
92     return ScheduleTester()
93
94 @pytest.fixture
95 def my_expected_schedule(self):
96     """Фикстура с МОИМ ожидаемым расписанием"""
97     return {
98         "Понедельник": [
99             {
100                 "time": "9:00-10:30",
101                 "subject": "Математический анализ",
102                 "teacher": "Иванов И.И.",
103                 "room": "101"
104             },
105             {
106                 "time": "10:45-12:15",
107                 "subject": "Физика",
108                 "teacher": "Петров П.П.",
109                 "room": "203"
110             }
111         ],
112         "Вторник": [
113             {
114                 "time": "9:00-10:30",
115                 "subject": "Программирование",
116                 "teacher": "Сидоров С.С.",

```

Error: The following package(s) are not supported:
selenium

```

main x
144 }
145
146 @pytest.fixture
147 def different_schedule(self):
148     """Фикстура с ДРУГИМ расписанием (для демонстрации провала тестов)"""
149     return {
150         "Понедельник": [
151             {
152                 "time": "9:00-10:30",
153                 "subject": "ХИМИЯ", # Измененный предмет
154                 "teacher": "Иванов И.И.",
155                 "room": "101"
156             }
157         ],
158         "Вторник": [] # Пустой день
159     }
160
161 # БАЗОВЫЕ ТЕСТЫ
162 def test_api_availability(self, schedule_tester):
163     """Базовый тест: доступность API"""
164     schedule = schedule_tester.get_current_schedule_from_api()
165     assert schedule is not None, "API недоступно"
166
167 def test_schedule_structure(self, schedule_tester, my_expected_schedule):
168     """Базовый тест: структура расписания"""
169     schedule_tester.load_expected_schedule(my_expected_schedule)
170
171     # Проверяем, что ожидаемое расписание имеет правильную структуру
172     assert isinstance(my_expected_schedule, dict)
173     assert "Понедельник" in my_expected_schedule
174     assert isinstance(my_expected_schedule["Понедельник"], list)
175
176 def test_lesson_format(self, schedule_tester, my_expected_schedule):
177     """Базовый тест: формат урока"""
178     monday_lessons = my_expected_schedule["Понедельник"]
179     first_lesson = monday_lessons[0]
180
181     required_fields = ["time", "subject", "teacher", "room"]

```

Error: The following package(s) are not supported:
selenium

```

187 - @pytest.mark.parametrize("day,expected_lessons_count", [
188 -     ("Понедельник", 2),
189 -     ("Вторник", 1),
190 -     ("Среда", 1),
191 -     ("Четверг", 1),
192 -     ("Пятница", 1),
193 - ])
194 - def test_days_lesson_count(self, schedule_tester, my_expected_schedule, day, expected_lessons_count):
195 -     """Параметризованный тест: количество уроков в каждый день"""
196 -     schedule_tester.load_expected_schedule(my_expected_schedule)
197 -     actual_lessons_count = len(my_expected_schedule[day])
198 -     assert actual_lessons_count == expected_lessons_count, \
199 -         f"В {day} ожидалось {expected_lessons_count} уроков, но найдено {actual_lessons_count}"
200 -
201 - @pytest.mark.parametrize("day,lesson_index,expected_subject", [
202 -     ("Понедельник", 0, "Математический анализ"),
203 -     ("Понедельник", 1, "Физика"),
204 -     ("Вторник", 0, "Программирование"),
205 -     ("Среда", 0, "Базы данных"),
206 - ])
207 - def test_specific_lessons(self, schedule_tester, my_expected_schedule, day, lesson_index, expected_subje
208 -     """Параметризованный тест: проверка конкретных уроков"""
209 -     schedule_tester.load_expected_schedule(my_expected_schedule)
210 -     actual_subject = my_expected_schedule[day][lesson_index]["subject"]
211 -     assert actual_subject == expected_subject, \
212 -         f"В {day} урок {lesson_index}: ожидался '{expected_subject}', но получен '{actual_subject}'"
213 -
214 - @pytest.mark.parametrize("field,value", [
215 -     ("time", "9:00-10:30"),
216 -     ("teacher", "Иванов И.И."),
217 -     ("room", "101"),
218 - ])
219 - def test_lesson_fields(self, schedule_tester, my_expected_schedule, field, value):
220 -     """Параметризованный тест: проверка полей урока"""
221 -     schedule_tester.load_expected_schedule(my_expected_schedule)
222 -     monday_first_lesson = my_expected_schedule["Понедельник"][0]
223 -     assert monday_first_lesson[field] == value, \
224 -         f"Поле {field}: ожидалось '{value}', но получено '{monday_first_lesson[field]}'"
225 -
226 -
227 -
228 -
229 -
230 -
231 -
232 -
233 -
234 -
235 -
236 -
237 -
238 -
239 -
240 -
241 -
242 -
243 -
244 -
245 -
246 -
247 -
248 -
249 -
250 -
251 -
252 -
253 -
254 -
255 -
256 -
257 -
258 -
259 -
260 -
261 -

```

Error: The following package(s) are not supported:
selenium

```

224 -     f"Поле {field}: ожидалось '{value}', но получено '{monday_first_lesson[field]}'"
225 -
226 - # ТЕСТИРОВАНИЕ ИСКЛЮЧЕНИЙ
227 - def test_empty_schedule_exception(self, schedule_tester):
228 -     """Тест исключения: пустое расписание"""
229 -     with pytest.raises(AssertionError):
230 -         empty_schedule = {}
231 -         schedule_tester.load_expected_schedule(empty_schedule)
232 -         assert empty_schedule, "Расписание не должно быть пустым"
233 -
234 - def test_invalid_schedule_structure(self, schedule_tester):
235 -     """Тест исключения: некорректная структура расписания"""
236 -     invalid_schedule = {"Понедельник": "не список"}
237 -     schedule_tester.load_expected_schedule(invalid_schedule)
238 -
239 -     with pytest.raises(AssertionError):
240 -         assert isinstance(invalid_schedule["Понедельник"], list), \
241 -             "Уроки должны быть представлены списком"
242 -
243 - # ТЕСТ С SELENIUM (интеграционный)
244 - @pytest.mark.selenium
245 - @pytest.mark.integration
246 - def test_website_schedule_display(self, chrome_driver, schedule_tester):
247 -     """Тест с Selenium: отображение расписания на сайте"""
248 -     try:
249 -         chrome_driver.get("https://api.ntnu.ru")
250 -
251 -         # Здесь нужно адаптировать селекторы под реальную структуру сайта
252 -         # Это пример - замените на реальные селекторы
253 -
254 -         # Ждем загрузки страницы
255 -         WebDriverWait(chrome_driver, 10).until(
256 -             EC.presence_of_element_located((By.TAG_NAME, "body")))
257 -
258 -         # Проверяем, что страница загружена
259 -         assert "ННТУ" in chrome_driver.title or "ntnu" in chrome_driver.title.lower()
260 -
261 -
262 -

```

Error: The following package(s) are not supported:
selenium

Ln: 11, Col: 1

main x

Ln: 11, Col: 1

261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336

Делаем скриншот для отладки

chrome_driver.save_screenshot("schedule_page.png")

except Exception as e:

pytest.fail(f"Ошибка при тестировании веб-сайта: {e}")

ТЕСТЫ С МЕТКАМИ

@pytest.mark.my_schedule

def test_my_schedule_matches_expected(self, schedule_tester, my_expected_schedule):

"""Тест с меткой: МОЕ расписание соответствует ожидаемому"""

schedule_tester.load_expected_schedule(my_expected_schedule)

Проверяем ключевые аспекты моего расписания

assert "Программирование" in str(my_expected_schedule), "Должен быть урок программирования"

assert "Базы данных" in str(my_expected_schedule), "Должен быть урок баз данных"

Проверяем конкретные дни

assert len(my_expected_schedule["Понедельник"]) == 2

assert len(my_expected_schedule["Вторник"]) == 1

@pytest.mark.validation

def test_schedule_validation(self, schedule_tester, my_expected_schedule):

"""Тест с меткой: валидация расписания"""

schedule_tester.load_expected_schedule(my_expected_schedule)

for day, lessons in my_expected_schedule.items():

for lesson in lessons:

Проверяем формат времени

time_str = lesson["time"]

assert ":" in time_str and "-" in time_str, f"Некорректный формат времени: {time_str}"

Проверяем, что все поля заполнены

for field, value in lesson.items():

assert value, f"Пустое поле {field} в {day}"

ДЕМОНСТРАЦИЯ ПРОВАЛА ТЕСТОВ ПРИ ИЗМЕНЕНИИ РАСПИСАНИЯ

def test_my_schedule_vs_different_schedule_fails(self, schedule_tester, my_expected_schedule, different_

"""Демонстрация: тесты не проходят при изменении расписания"""

Error: The following package(s) are not supported:
selenium

main x

Ln: 11, Col: 1

298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336

def test_my_schedule_vs_different_schedule_fails(self, schedule_tester, my_expected_schedule, different_

"""Демонстрация: тесты не проходят при изменении расписания"""

schedule_tester.load_expected_schedule(my_expected_schedule)

Этот тест должен провалиться при использовании different_schedule

Давайте протестируем оба сценария:

Сценарий 1: Мое расписание - тесты проходят

monday_lessons = my_expected_schedule["Понедельник"]

assert len(monday_lessons) == 2

assert monday_lessons[0]["subject"] == "Математический анализ"

Сценарий 2: Другое расписание - тесты НЕ проходят

Раскомментируйте для демонстрации провала:

"""

schedule_tester.load_expected_schedule(different_schedule)

monday_lessons_diff = different_schedule["Понедельник"]

Эти утверждения провалятся:

assert len(monday_lessons_diff) == 2 # Провал: ожидается 2, но получено 1

assert monday_lessons_diff[0]["subject"] == "Математический анализ" # Провал: получено "ХИМИЯ"

"""

ДОПОЛНИТЕЛЬНЫЙ КЛАСС ДЛЯ ДЕМОНСТРАЦИИ ПРОВАЛА ТЕСТОВ

class TestScheduleFailureDemo:

"""Класс для демонстрации провала тестов при измененном расписании"""

@pytest.fixture

def changed_schedule(self):

"""Расписание с изменениями, которые вызовут провал тестов"""

return {

"Понедельник": [

{

"time": "9:00-10:30",

"subject": "ХИМИЯ", # Изменено!

"teacher": "Иванов И.И.",

"room": "101"

},

],

Error: The following package(s) are not supported:
selenium

```
main x
333     "subject": "ХИМИЯ", # Изменено!
334     "teacher": "Иванов И.И.",
335     "room": "101"
336 },
337     # Удален второй урок
338 ],
339     "Вторник": [], # Пустой день вместо программирования
340     "Среда": [
341         {
342             "time": "13:00-14:30",
343             "subject": "ИСТОРИЯ", # Изменено!
344             "teacher": "Кузнецова К.К.",
345             "room": "412"
346         }
347     ]
348 }
349
350 @pytest.mark.failure_demo
351 def test_demonstrate_failure(self, changed_schedule):
352     """Демонстрация провала тестов при измененном расписании"""
353     # Все эти тесты провалятся:
354
355     # Провал: ожидается 2 урока в понедельник
356     assert len(changed_schedule["Понедельник"]) == 2
357
358     # Провал: ожидается "Математический анализ"
359     assert changed_schedule["Понедельник"][0]["subject"] == "Математический анализ"
360
361     # Провал: ожидается непустой вторник
362     assert len(changed_schedule["Вторник"]) > 0
363
364     # Провал: ожидается "Базы данных" в среду
365     assert changed_schedule["Среда"][0]["subject"] == "Базы данных"
366
367
368 if __name__ == "__main__":
369     # Запуск тестов
370     pytest.main(["-v", "--tb=short"])

Ln: 11. Col: 1
```

Error: The following package(s) are not supported:
selenium