

## Лабораторная работа №4

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

### Код задания

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time
import pytest

def test_announcements_items_count():
    # Автоматическая загрузка и установка правильной версии ChromeDriver
    service = Service(ChromeDriverManager().install())
    driver = webdriver.Chrome(service=service)

    try:
        # Установка неявного ожидания 20 секунд
        driver.implicitly_wait(20)

        # Открыть веб-страницу
        driver.get('https://api.nntu.ru/')

        # Ожидание загрузки страницы
        wait = WebDriverWait(driver, 20)

        # 0. Поиск и нажатие на кнопку "СОГЛАСЕН" в cookie-баннере
        try:
            # Ждем появления cookie-баннера
            cookie_banner = wait.until(EC.presence_of_element_located((By.ID,
"cookieBar"))))
            print("Найден cookie-баннер")

            # Ищем кнопку "СОГЛАСЕН" внутри баннера
            agree_button = cookie_banner.find_element(
                By.XPATH, "//*[@button[contains(text(), 'СОГЛАСЕН')]]")

            # Нажимаем на кнопку
            agree_button.click()
            print("Нажата кнопка 'СОГЛАСЕН' в cookie-баннере")

            # Ждем исчезновения баннера
            wait.until(EC.invisibility_of_element_located((By.ID,
"cookieBar"))))
            print("Cookie-баннер скрыт")

        except Exception as e:
            print(f"Cookie-баннер не найден или не требуется принятие: {e}")

        # 1. Поиск и наведение на элемент "Жизнь в АПИ НГТУ"
        life_element = wait.until(EC.element_to_be_clickable(
            (By.XPATH, "//a[@class='nav-link' and contains(text(), 'Жизнь в
АПИ НГТУ')]]"))
    )
```

```

actions = ActionChains(driver)
actions.move_to_element(life_element).perform()
print("Курсор наведен на элемент 'Жизнь в АПИ НГТУ'")
time.sleep(1)

# 2. Поиск ссылки "События и новости" в выпадающем меню
news_element = wait.until(EC.element_to_be_clickable(
    (By.XPATH, "//a[contains(text(), 'События и новости')]")
))

# Наведение курсора на "События и новости"
actions.move_to_element(news_element).perform()
print("Курсор наведен на элемент 'События и новости'")

# Пауза 1 секунда
time.sleep(1)

# 3. Клик и переход по ссылке
news_element.click()
print("Выполнен переход по ссылке 'События и новости'")

# Ожидание загрузки новой страницы
wait.until(lambda driver: any([
    "news" in driver.current_url,
    "events" in driver.current_url,
    "sobytiya" in driver.current_url,
    "novosti" in driver.current_url
]))
print(f"Текущий URL: {driver.current_url}")

# 4. Поиск элемента с классом rubric-nav
rubric_nav =
wait.until(EC.presence_of_element_located((By.CLASS_NAME, "rubric-nav")))
print("Найден элемент с классом 'rubric-nav'")

# 5. Поиск всех элементов рубрик внутри rubric-nav
rubric_items = rubric_nav.find_elements(By.CLASS_NAME, "rubric-item")
print(f"Найдено элементов рубрик: {len(rubric_items)}")

# 6. Проход курсором по всем элементам рубрик с задержкой 0.5 секунды
actions = ActionChains(driver)
for i, item in enumerate(rubric_items):
    actions.move_to_element(item).pause(0.5).perform()
    print(f"Курсор наведен на рубрику {i+1}: '{item.text}'")

# 7. Поиск элемента "Объявления"
announcement_element = wait.until(EC.element_to_be_clickable(
    (By.XPATH, "//li[contains(@class, 'rubric-
item')]//*[contains(text(), 'Объявления')]")
))

# 8. Наведение на элемент "Объявления"
actions.move_to_element(announcement_element).perform()
print("Курсор наведен на элемент 'Объявления'")

# Пауза 1 секунда
time.sleep(1)

# 9. Переход по элементу "Объявления"
announcement_element.click()
print("Выполнен переход по элементу 'Объявления'")

# Ожидание загрузки страницы объявлений

```

```

wait.until(lambda driver: any([
    "announcements" in driver.current_url,
    "obyavleniya" in driver.current_url,
    "ads" in driver.current_url,
    "объявлен" in driver.current_url.lower()
]))
print(f"Текущий URL страницы объявлений: {driver.current_url}")

# 10. Поиск контейнера events-items
events_container = wait.until(EC.presence_of_element_located(
    (By.CLASS_NAME, "events-items")
))
print("Найден контейнер events-items")

# 11. Поиск всех элементов с классом item внутри контейнера
item_elements = events_container.find_elements(By.CLASS_NAME, "item")
items_count = len(item_elements)
print(f"Найдено элементов с классом 'item': {items_count}")

# 12. Проверка с помощью pytest assert
assert items_count == 9, f"Ожидалось 9 элементов, но найдено {items_count}"

# 13. Дополнительная информация для отладки
print("Список найденных элементов:")
for i, item in enumerate(item_elements, 1):
    # Попробуем получить текст элемента, если возможно
    try:
        item_text = item.text[:50] + "..." if len(item.text) > 50
    except:
        item_text = "Не удалось получить текст"
    print(f"Элемент {i}: {item_text}")

print("✓ Тест пройден: найдено ровно 9 элементов с классом 'item'")

# Дополнительная пауза для просмотра результата
time.sleep(2)

finally:
    # Закрыть драйвер после использования
    driver.quit()

# Запуск теста
if __name__ == "__main__":
    test_announcements_items_count()

```

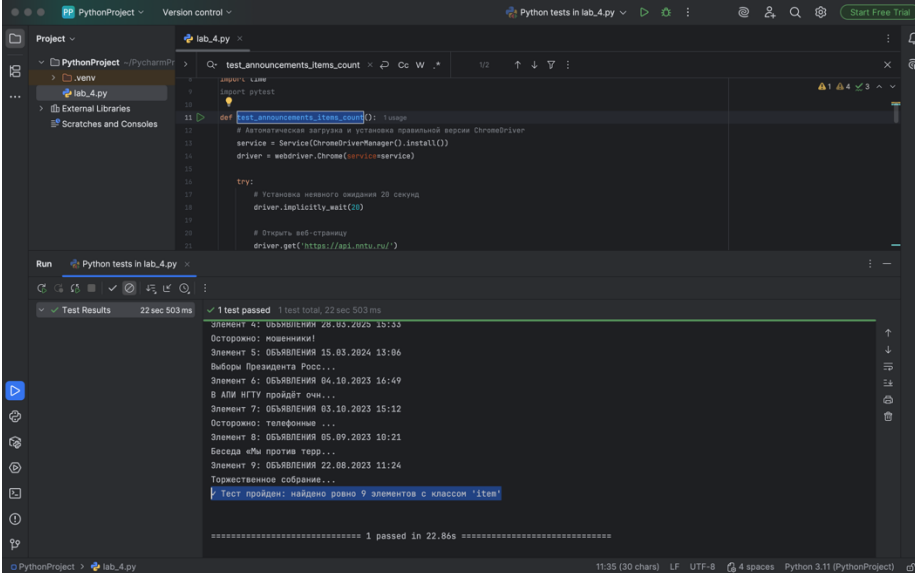
## Описание

Открывается страница <https://api.nntu.ru/>. Соглашаемся с куки банером. Наводим курсор на элемент с классом «nav-link» с текстом «Жизнь в АПИ НГТУ». Ждем открытия меню. Ищем в выпадающем меню пункт «События и новости». Переходим на новую страницу.

Ищем элемент с классом «rubric-nav». Проходим курсором по всем элементам рубрик(элементы с классом «rubric-item»). Ищем элемент в списке навигации с текстом «Объявления» Переходим на страницу с объявлениями.

С помощью **pytest** проверяем количество объявлений на странице.

Их должно быть 9. **Тест пройден!**



The screenshot shows the PyCharm IDE interface. The top pane displays the code for a test file named `test_announcements_items_count.py`. The code uses Selenium with WebDriver to interact with a web page. The test function `test_announcements_items_count` is highlighted. The bottom pane shows the 'Run' output, indicating that the test passed successfully. The output text lists 9 items found on the page, each with a timestamp, and confirms that the test passed because exactly 9 items were found with the class 'item'.

```
import time
import pytest

def test_announcements_items_count():
    """
    # Автоматическая загрузка и установка правильной версии ChromeDriver
    service = Service(ChromeDriverManager().install())
    driver = webdriver.Chrome(service=service)

    try:
        # Установка неявного ожидания 20 секунд
        driver.implicitly_wait(20)

        # Открыть веб-страницу
        driver.get('https://api.mstu.ru/')

    except Exception as e:
        print(e)
    finally:
        driver.quit()
```

Run Python tests in lab\_4.py

✓ 1 test passed 1 test total, 22 sec 503 ms

Элемент 4: ОБЪЯВЛЕНИЯ 20.03.2023 15:33  
Осторожно: мошенники!  
Элемент 5: ОБЪЯВЛЕНИЯ 15.03.2024 13:06  
Выборы Президента Росс...  
Элемент 6: ОБЪЯВЛЕНИЯ 04.10.2023 16:49  
В АПИ НГТУ пройдут он...  
Элемент 7: ОБЪЯВЛЕНИЯ 03.10.2023 15:12  
Осторожно: телефонные ...  
Элемент 8: ОБЪЯВЛЕНИЯ 05.09.2023 10:21  
Беседа «Мы против терр...  
Элемент 9: ОБЪЯВЛЕНИЯ 22.08.2023 11:24  
Торжественное собрание...

✓ Тест пройден: найдено ровно 9 элементов с классом 'item'

===== 1 passed in 22.86s =====