

Задали корректное расписание константой. Запуск теста:

The screenshot shows the VS Code interface with the file explorer on the left displaying a project structure including `main.py` and `tests.py`. The main editor displays the code for `main.py`, which includes a `CORRECT_SCHEDULE` constant and a `TestScheduleParser` class with a `test_get_schedule_returns_correct_data` method. The terminal at the bottom shows the output of running `python main.py`, indicating that the test passed successfully. The output includes a warning about `pytest.mark.basic` and a message from the Google Assistant API.

```
1 import pytest
2 from ScheduleParser import ScheduleParser
3 import subprocess
4 import sys
5
6 CORRECT_SCHEDULE = [
7     ["2 пара / 10:10-11:40", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
8     ["3 пара / 12:10-13:40", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
9     ["4 пара / 13:50-15:20", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
10    ["5 пара / 15:30-17:00", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
11    ["1 пара вечер / 17:10-19:00", "Теория цифровой обработки сигналов / Лек.", "Абаимов Анатолий Вячеславович", "317", "", "Четная"],
12    ["2 пара вечер / 19:10-20:40", "Теория цифровой обработки сигналов / Лек.", "Абаимов Анатолий Вячеславович", "317", "", "Четная"]
13 ]
14
15 class TestScheduleParser:
16
17     @pytest.mark.basic
18     def test_get_schedule_returns_correct_data(self):
19         """
20         Тест проверяет, что функция get_schedule() возвращает корректные данные расписания.
21         """
22         # Act - вызываем тестируемую функцию
23         response = ScheduleParser.get_schedule()
24         assert response is not None
25         assert response == CORRECT_SCHEDULE
26
27 if __name__ == "__main__":
28     pytest.main([__file__, "-v"])
```

main.py:10
c:\Users\Andrew\Desktop\Тестирование\VP4\main.py:10: PytestUnknownMarkWarning: Unknown pytest.mark.basic - is this a typo? You can register custom marks to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
@pytest.mark.basic
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
1 passed, 1 warning in 17.58s
PS C:\Users\Andrew\Desktop\Тестирование\VP4> [3340:8.6919/163443.984:ERROR:google_api\grpcengine\registration_request.cc:291] Registration response error message: DEPRECATED_ENDPOINT

Изменили расписание в константе на некорректное. Запуск теста:

The screenshot shows the VS Code interface with the file explorer on the left displaying a project structure including `main.py` and `tests.py`. The main editor displays the code for `main.py`, which includes a `CORRECT_SCHEDULE` constant and a `TestScheduleParser` class with a `test_get_schedule_returns_correct_data` method. The terminal at the bottom shows the output of running `python main.py`, indicating that the test failed. The output includes a warning about `pytest.mark.basic` and a message from the Google Assistant API.

```
1 import pytest
2 from ScheduleParser import ScheduleParser
3 import subprocess
4 import sys
5
6 CORRECT_SCHEDULE = [
7     ["2 пара / 10:10-11:40", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
8     ["3 пара / 12:10-13:40", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
9     ["4 пара / 13:50-15:20", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
10    ["5 пара / 15:30-17:00", "Анализ больших данных / Лек.", "Ригов Антон Владимирович", "320", "", "Четная"],
11    ["1 пара вечер / 17:10-19:00", "Теория цифровой обработки сигналов / Лек.", "Абаимов Анатолий Вячеславович", "317", "", "Четная"],
12    ["2 пара вечер / 19:10-20:40", "Теория цифровой обработки сигналов / Лек.", "Абаимов Анатолий Вячеславович", "317", "", "Четная"]
13 ]
14
15 class TestScheduleParser:
16
17     @pytest.mark.basic
18     def test_get_schedule_returns_correct_data(self):
19         """
20         Тест проверяет, что функция get_schedule() возвращает корректные данные расписания.
21         """
22         # Act - вызываем тестируемую функцию
23         response = ScheduleParser.get_schedule()
24         assert response is not None
25         assert response == CORRECT_SCHEDULE
26
27 if __name__ == "__main__":
28     pytest.main([__file__, "-v"])
```

main.py:21: AssertionError
main.py:13
c:\Users\Andrew\Desktop\Тестирование\VP4\main.py:13: PytestUnknownMarkWarning: Unknown pytest.mark.basic - is this a typo? You can register custom marks to avoid this warning - for details, see https://docs.pytest.org/en/stable/how-to/mark.html
@pytest.mark.basic
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
short test summary info
FAILED main.py::TestScheduleParser::test_get_schedule_returns_correct_data - AssertionError: assert [['2 пара / 1...', 'Четная']] == [['2 пара / 1...', 'Четная']]
1 failed, 1 warning in 19.27s
PS C:\Users\Andrew\Desktop\Тестирование\VP4>

Код парсера:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By
from time import *

class ScheduleParser:

    def parse_table_element(table_element):
        """Парсит таблицу из WebElement"""
        data = []

        # Важно: используем find_elements (с s!) для поиска внутри элемента
        rows = table_element.find_elements(By.TAG_NAME, "tr")

        for row in rows:
            # Ищем ячейки внутри каждой строки
            cells = row.find_elements(By.TAG_NAME, "td")
            if cells: # Только строки с данными
                row_data = [cell.text for cell in cells]
                data.append(row_data)
        return data

    def get_schedule():
        # Настройка опций Chrome
        chrome_options = Options()
        chrome_options.add_experimental_option("detach", True)
        driver_path = r'C:\Users\Andrew\Desktop\Тестирование\ЛР4\chromedriver-win64\chromedriver.exe'
        service = Service(executable_path=driver_path)

        driver = webdriver.Chrome(service=service, options=chrome_options)
        driver.get('https://api.ntu.ru/raspisanie')
        sleep(5)
        list = Select(driver.find_element(By.XPATH,
            "/html/body/div[3]/div[3]/div/div/div/div[2]/div[2]/select"))
        list.select_by_value("3")
        sleep(1)
        group_list =
        Select(driver.find_element(By.XPATH, "/html/body/div[3]/div[3]/div/div/div/div[3]/div[2]/select"))
        group_list.select_by_value("783")
        sleep(3)
        schedule =
        driver.find_element(By.XPATH, "/html/body/div[3]/div[3]/div/div/div/div[8]/div[2]/table[1]")
        return ScheduleParser.parse_table_element(schedule)
```