# Page Object

Теория:

<https://kreisfahrer.gitbooks.io/selenium-webdriver/content/page_object_pattern_arhitektura_testovogo_proekta/ispolzovanie_patterna_page_object.html>

<https://webdriver.io/docs/pageobjects.html> <<< очень хороший пример, ориентируйтесь на него

Практика:

Переписать имеющийся Google Test с использованием **Page Object шаблона**.

- зайти на сайт google.com

- выполнить поиск по слову “iTechArt”

- на первых двух страницах результатов **проверить**, что слово “iTechArt” (регистр не важен) присутствует в каждом из 10 блоков результатов поиска

- **проверить**, что количество результатов поиска (строка вверху над результатами) больше X (задаётся параметром, см. ниже)

- вывести в консоль количество результатов и время поиска (то, что рядом с количеством результатов)

Требования:

- созданы page object классы для: стартовой страницы и страницы с результатами

- в методах page object классов:

- вся логика работы с элементами (ожидание, ввод текста, клик и т.д.)

- непосредственно в файле теста:

- обращение к методам page object классов

- логика по выбору браузера и его запуску

- проверки (expect)

Критерии оценки

|  |  |
| --- | --- |
| Файлы, не относящиеся к коду, добавлены в .gitignore и отсутствуют в репозитории (node\_modules, .idea и т.д.) | 1 |
| Вся подготовка (установка пакетов, драйверов) осуществляется с помощью npm. Не должно быть необходимости скачивать и устанавливать драйвера для браузера вручную, устанавливать пакеты глобально, добавлять в path и т.д. | 1 |
| Тест запускается командой `npm test` | 0.5 |
| Проверки выполняются с помощью с библиотеки jasmine: expect(…) | 1 |
| Выполнены обе проверки | 2 |
| Каждый из шагов теста оформлен в отдельный `it` блок с информативным заголовком | 0.5 |
| Параметризация:  Слово для поиска и количество результатов для сравнения заданы в отдельном файле с данными testData.js  Параметры, которые не относятся конкретно к данному тесту, вынесены в файл конфигурации | 1 |
| По умолчанию тест запускается на браузере chrome, имеется возможность параметром задать браузер: npm test -- --browser=firefox | 1 |
| Отсутствуют дублирование кода и орфографические ошибки | 0.5 |
| Наличие Readme (описано, как настроить и запускать тест) | 1 |
| По окончанию теста браузер закрывается автоматически вне зависимости от результатов | 0.5 |
| Тест не запускается, падает раньше времени, падает по таймауту, выбрасывает exception, проверить его работоспособность не представляется возможным | -3 |
| Непосредственно с самом файле теста присутствует логика работы с элементами и selenium-ом | -2 |
| В классах Page Object присутствует логика теста / проверки expect | -2 |
| Использован expect(…).toBe(…) вместо более информативных  expect(…).toContain(…)  expect(…).toBeGreaterThan(…) | -1 |
| Хардкод в локаторах:  - локаторы с индексами //div[1] | -1 |
| Локаторы элементов описаны внутри методов page object там, где этого можно избежать | -1 |