

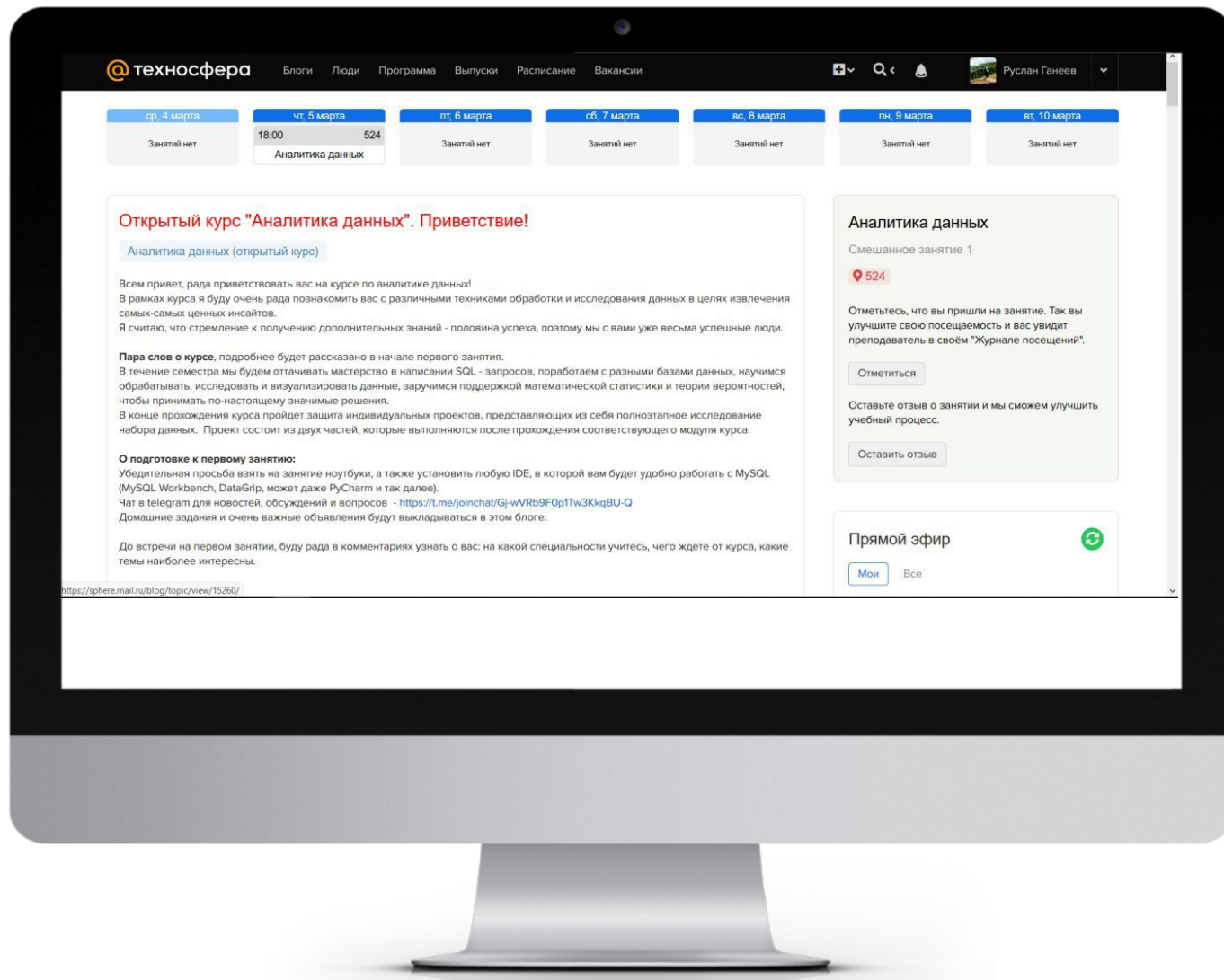
Образовательные проекты

mail.ru
group


Автоматизация тестирования на Python

Константин Ермаков





Не забудьте
отметиться
на портале



Тестирование UI: Page Object

1. Суть паттерна
2. В чем преимущество
3. Примеры
4. Преобразуем код к новому виду

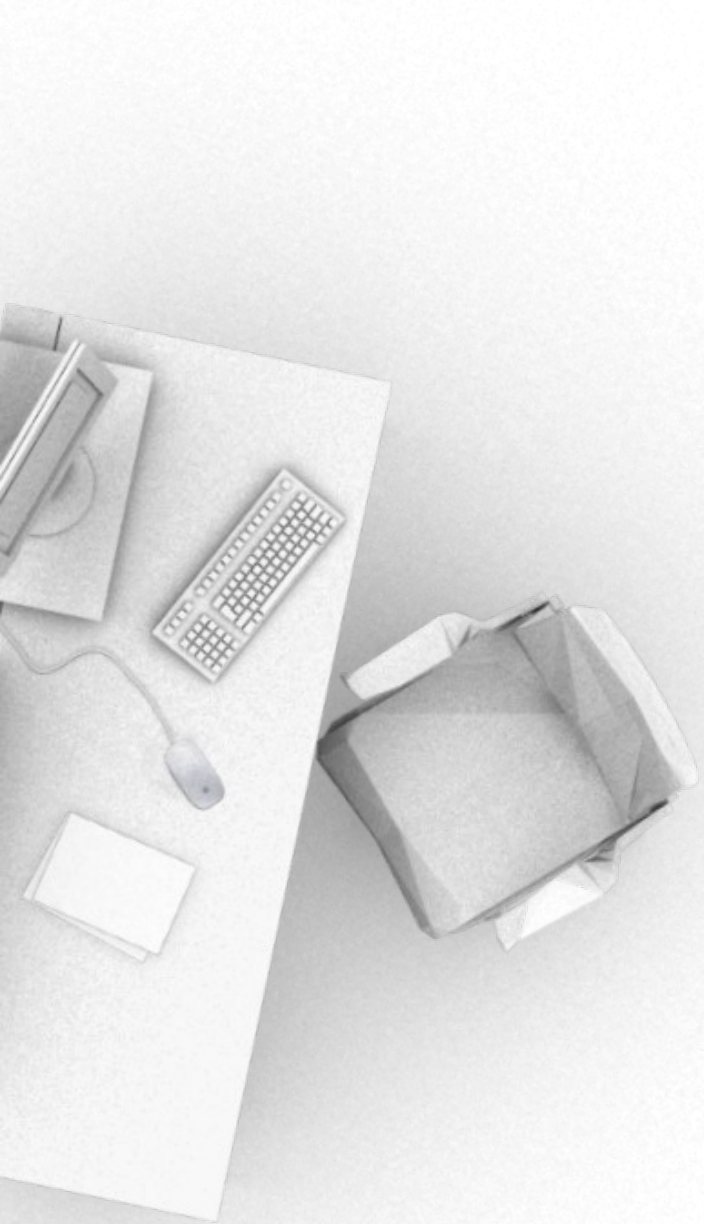


Page Object:
Суть паттерна

Что такое Page Object?

Это популярный паттерн, который является по сути стандартом в автоматизации тестирования UI веб-продуктов.

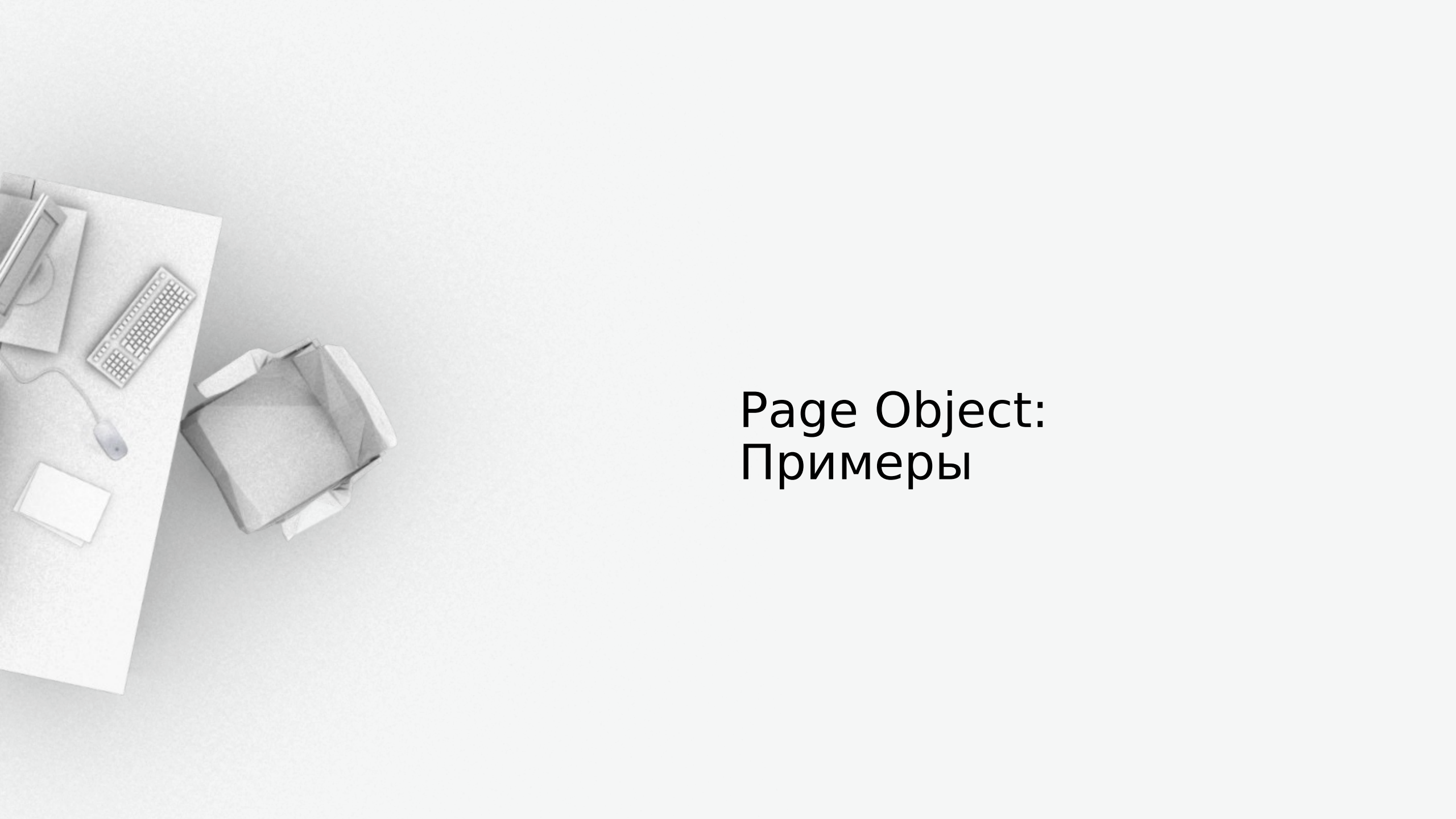
Основная идея состоит в том, чтобы разделить логику тестов от реализации.



Page Object:
В чем преимущество

Преимущества Page Object

- Разделение кода тестов и их реализации
- Уменьшение дублирования кода
- Увеличение переиспользуемости кода
- Увеличение читаемости кода



Page Object: Примеры

Selenium WebDriver: pytest example

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

def test():
    driver = webdriver.Chrome()
    driver.get("http://www.python.org")
    assert "Python" in driver.title

    elem = driver.find_element_by_name("q")
    elem.send_keys("pycon")
    elem.send_keys(Keys.RETURN)
    assert "No results found." not in driver.page_source

    driver.close()
```

Selenium WebDriver: pytest example with PO

```
import pytest
from base_tests.base import BaseCase

class TestOne(BaseCase):
    @pytest.mark.parametrize(
        'query',
        [
            pytest.param(
                'pycon'
            ),
            pytest.param(
                'python'
            ),
        ]
    )
    def test_search(self, query):
        self.base_page.search(query)
        assert "No results found." not in self.driver.page_source
```

Selenium WebDriver: pytest example Base Page

```
from selenium.common.exceptions import StaleElementReferenceException
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait

from ui.locators.pages_locators import BasePageLocators

CLICK_RETRY = 3

class BasePage(object):

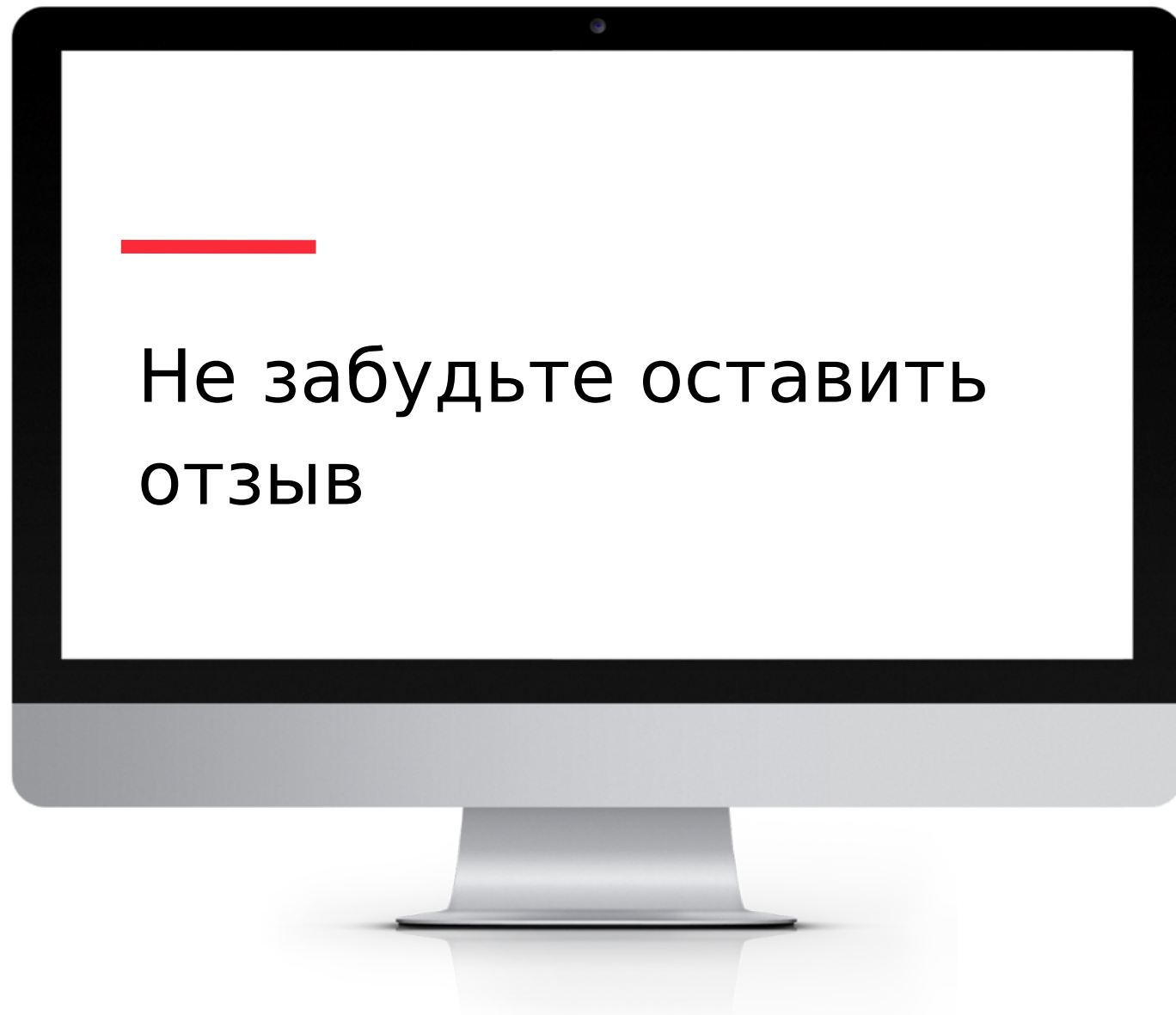
    locators = BasePageLocators()

    def __init__(self, driver):
        self.driver = driver

    def find(self, locator, timeout=None):
        return self.wait(timeout).until(EC.presence_of_element_located(locator))

    def wait(self, timeout=None):
        if timeout is None:
            timeout = 5
        return WebDriverWait(self.driver, timeout=timeout)

    def search(self, query):
        search = self.find(self.locators.QUERY_LOCATOR)
        search.clear()
        search.send_keys(query)
        self.click(self.locators.GO_LOCATOR)
```



—

Не забудьте оставить
ОТЗЫВ

**СПАСИБО
ЗА ВНИМАНИЕ**

