**Propmpt**

przygotuj bibliotękę w pythonie bazując na wzorcu projektowym page object pattern dla strony <https://demoqa.com/>

**ChatGPT powiedział:**

ChatGPT

Tworzenie biblioteki w Pythonie bazującej na **Page Object Pattern** dla strony [demoqa.com](https://demoqa.com/) wymaga organizacji kodu, która oddziela logikę testów od logiki interakcji z elementami strony. Oto propozycja takiej biblioteki:

**Struktura projektu**

bash

Skopiuj kod

/demoqa\_page\_objects

\_\_init\_\_.py

base\_page.py

home\_page.py

elements\_page.py

/tests

test\_demoqa.py

requirements.txt

**1. Plik base\_page.py**

Klasa bazowa, która dostarcza ogólne metody dla każdej strony.

python

Skopiuj kod

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

class BasePage:

def \_\_init\_\_(self, driver):

self.driver = driver

self.wait = WebDriverWait(driver, 10)

def open(self, url):

"""Otwiera stronę o podanym URL."""

self.driver.get(url)

def find\_element(self, locator):

"""Znajduje pojedynczy element."""

return self.wait.until(EC.visibility\_of\_element\_located(locator))

def click\_element(self, locator):

"""Kliknięcie w element."""

element = self.find\_element(locator)

element.click()

def enter\_text(self, locator, text):

"""Wpisuje tekst do pola."""

element = self.find\_element(locator)

element.clear()

element.send\_keys(text)

**2. Plik home\_page.py**

Page Object dla strony głównej.

python

Skopiuj kod

from demoqa\_page\_objects.base\_page import BasePage

from selenium.webdriver.common.by import By

class HomePage(BasePage):

ELEMENTS\_CARD = (By.XPATH, "//h5[text()='Elements']")

def navigate\_to\_elements(self):

"""Przejście do sekcji 'Elements'."""

self.click\_element(self.ELEMENTS\_CARD)

**3. Plik elements\_page.py**

Page Object dla sekcji "Elements".

python

Skopiuj kod

from demoqa\_page\_objects.base\_page import BasePage

from selenium.webdriver.common.by import By

class ElementsPage(BasePage):

TEXT\_BOX\_MENU = (By.ID, "item-0")

FULL\_NAME\_FIELD = (By.ID, "userName")

EMAIL\_FIELD = (By.ID, "userEmail")

SUBMIT\_BUTTON = (By.ID, "submit")

OUTPUT\_TEXT = (By.ID, "output")

def open\_text\_box(self):

"""Przejście do Text Box."""

self.click\_element(self.TEXT\_BOX\_MENU)

def fill\_text\_box(self, name, email):

"""Wypełnienie formularza Text Box."""

self.enter\_text(self.FULL\_NAME\_FIELD, name)

self.enter\_text(self.EMAIL\_FIELD, email)

self.click\_element(self.SUBMIT\_BUTTON)

def get\_output\_text(self):

"""Pobiera tekst z sekcji wyników."""

return self.find\_element(self.OUTPUT\_TEXT).text

**4. Plik test\_demoqa.py**

Przykładowy test, który wykorzystuje Page Object Pattern.

python

Skopiuj kod

import unittest

from selenium import webdriver

from demoqa\_page\_objects.home\_page import HomePage

from demoqa\_page\_objects.elements\_page import ElementsPage

class DemoQATests(unittest.TestCase):

@classmethod

def setUpClass(cls):

cls.driver = webdriver.Chrome()

cls.driver.maximize\_window()

@classmethod

def tearDownClass(cls):

cls.driver.quit()

def test\_text\_box\_submission(self):

home\_page = HomePage(self.driver)

elements\_page = ElementsPage(self.driver)

home\_page.open("https://demoqa.com/")

home\_page.navigate\_to\_elements()

elements\_page.open\_text\_box()

elements\_page.fill\_text\_box("John Doe", "johndoe@example.com")

output\_text = elements\_page.get\_output\_text()

self.assertIn("John Doe", output\_text)

self.assertIn("johndoe@example.com", output\_text)

if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()

**5. Plik requirements.txt**

Zależności projektu.

makefile

Skopiuj kod

selenium==4.10.0

**Jak uruchomić?**

1. Zainstaluj wymagane pakiety:

bash

Skopiuj kod

pip install -r requirements.txt

1. Uruchom test:

bash

Skopiuj kod

python -m unittest tests/test\_demoqa.py

To podejście organizuje kod w modularny sposób, ułatwiając rozwój i utrzymanie testów. Możesz dodawać kolejne strony jako osobne klasy w katalogu page\_objects.