10. Java_QA (модель Page Object)

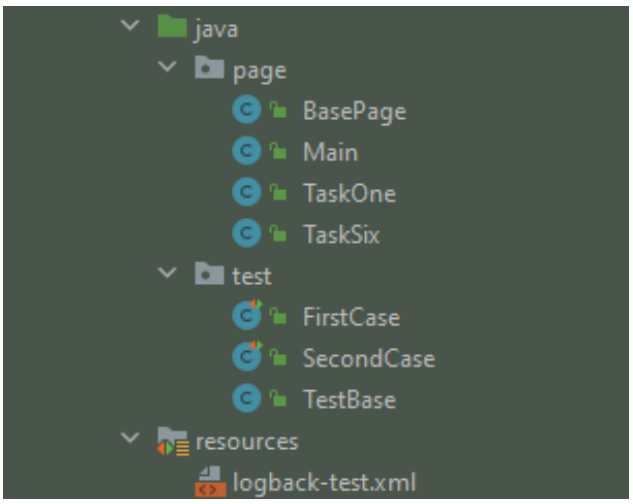
- Использование Паттерна PageObject при создании тестовых фреймворков

- Класс PageFactory

- Отключение логов

Преобразование ранее созданного проекта на основе Selenium, в модель Page Object.

Структура нового проекта:



page.Class BasePage: (конструктор веб-драйвера, вспомогательные методы)

```
public class BasePage {
    public WebDriver driver;
    public WebDriverWait wait;

    public BasePage(WebDriver driver) {
        this.driver = driver;
        wait = new WebDriverWait(driver, 15);
        return;
    }

    //wait wrapper method
    public void waitVisibility(By elementBy) {
        // expected conditions -- ожидаемые условия

wait.until(ExpectedConditions.visibilityOfAllElementsLocatedBy(elementBy));
    }

    //click method
    public void click(By elementBy) {
        waitVisibility(elementBy);
        driver.findElement(elementBy).click();
```

```
    }

    //isElementDisplayed
    public void isElementDisplayed(By elementBy) {
        waitVisibility(elementBy);
        assertTrue(driver.findElement(elementBy).isDisplayed());

    }

    public void waitForFilling(By elementBy) {
        wait.until(ExpectedConditions.elementToBeClickable(elementBy));
    }

    public void writeText(By elementBy, String text) {
        waitVisibility(elementBy);
        WebElement element = driver.findElement(elementBy);
        element.clear();
        element.sendKeys(text);

    }
```

page.Class Main: главный объект с основной страницей сайта, и методом перехода к другим страницам

```
public class Main extends BasePage {
    String SITE_URL = "https://buggy-testingcup.pgs-soft.com/";

    public Main(WebDriver driver) {
        super(driver);
    }

    public Main goTo() {
        driver.get(SITE_URL);
        return this;

    }

    public Main chooseTask(String taskNum) {
        click(By.xpath("//h2[.='Zadanie " + taskNum + "']"));
        return this;

    }
}
```

page.Class TaskOne: страница с заданием №1, и используемые на ней методы (проверка корректности выбранной страницы, и переход к элементу)

```
public class TaskOne extends BasePage {

    public TaskOne(WebDriver driver) {
        super(driver);
    }

    public void checkPageIsCorrect() {
        isElementDisplayed(By.xpath("//li[.='Zadanie 1']"));
    }
```

```
    public void scrollToElement(String element) {
        WebElement webElement = driver.findElement(By.xpath("//h4[text()='" +
element + "']"));
        Actions actions = new Actions(driver);
        actions.moveToElement(webElement);
        actions.perform();
    }
}
```

page.Class TaskSix: страница с заданием №6 с используемыми методами (ввод логина и пароля, нажатие кнопки логина, проверка логина, и полной загрузки страницы)

```
ublic class TaskSix extends BasePage {
    public TaskSix(WebDriver driver) {
        super(driver);
    }

    public TaskSix fillInLogin(String login) throws InterruptedException {
        writeText(By.id("LoginForm__username"), login);
        return this;
    }

    public TaskSix fillInPass(String pass) throws InterruptedException {
        writeText(By.name("LoginForm[_password]"), pass);

        return this;

    }

    public TaskSix logBtnClick() throws InterruptedException {
        click(By.cssSelector(".btn-default.btn"));
        return this;
    }

    public TaskSix isLoginCorrect() {
        isElementDisplayed(By.linkText("Pobierz plik"));
        return this;
    }

    //чтобы работало без SLEEP
    public TaskSix checkAllElementsOnPagePresent() {
        isElementDisplayed((By.id("LoginForm__username")));
        isElementDisplayed((By.name("LoginForm[_password]")));
        isElementDisplayed((By.cssSelector(".btn-default.btn")));
        return this;
    }
}
```

test.Class TestBase: создание экземпляров страниц, методы инициализации и закрытия драйвера

```
public class TestBase {
    WebDriver driver;
    public Main main;
    public TaskOne taskOne;
    public TaskSix taskSix;

    @BeforeEach
```

```
    public void start() {
        WebDriverManager.chromedriver().setup();
        ChromeOptions options = new ChromeOptions();
        options.addArguments("--test-type");
        ChromeDriverService chromeDriverService =
ChromeDriverService.createDefaultService();
        driver = new ChromeDriver(chromeDriverService, options);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
        //подтягивает элементы со страницы Main
        main = PageFactory.initElements(driver, Main.class);
        //подтягивает элементы со страницы TaskOne
        taskOne = PageFactory.initElements(driver, TaskOne.class);
        taskSix = PageFactory.initElements(driver, TaskSix.class);
    }

    @AfterEach
    public void finish() {
        if (driver != null)
            driver.quit();
    }
}
```

test.Class FirstCase: тестирование страницы №1, переход к заданию с главной страницы, проверка корректности страницы и переход к элементу

```
public class FirstCase extends TestBase {

    @Test
    public void firstTest() {
        main
                .goTo()
                .chooseTask("1");
        taskOne
                .checkPageIsCorrect();
        taskOne
                .scrollToElement("Kostka");
    }
}
```

test.Class SecondCase: тестирование страницы №6, проверка, что все элементы отображены, ввод логина и пароля, попытка авторизации и проверка данных

```
public class SecondCase extends TestBase {

    @Test
    public void SecondTest() throws InterruptedException {
        main.goTo()
                .chooseTask("6");
        taskSix.checkAllElementsOnPagePresent()
                .fillInLogin("tester")
                .fillInPass("123-xyz")
                .logBtnClick()
                .isLoginCorrect();
    }
}
```

Отключение логов

File logback-test.xml:

```
<!DOCTYPE configuration>
<configuration debug="false">
    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
            <pattern>%d [%thread] %-5level %logger{36} - %msg%n</pattern>
        </encoder>
    </appender>

    <root level="OFF">
        <appender-ref ref="STDOUT"/>
    </root>
</configuration>
```

Pom.xml:

```
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.2.3</version>
    <scope>test</scope>
</dependency>
```

Итоговый запуск тестов и результат: