

Iryna Ihnatsenka

Terrapizza.by

Задание

Terrapizza.by

(обязательное)

1. Зайти на домашнюю страницу сайта.
2. В разделе Пицца(-ы) добавить в Корзину пиццу Маргарита любого размера.
3. Перейти в Корзину.
4. Проверить, что пицца Маргарита есть в заказе.

(по желанию)

1. Зайти на домашнюю страницу сайта.
2. В разделе Пицца(-ы) добавить в Корзину пиццу Маргарита любого размера.
3. В разделе Напитки (Бар) добавить в Корзину любой напиток.
4. Перейти в Корзину.
5. Проверить, что пицца Маргарита есть в заказе.
6. Проверить, что выбранный напиток есть в заказе.

Технические требования

Terrapizza.by

- Java 8
- проект Maven (pom.xml)
- Junit5
- Webdriver, PageFactory, xpath
- Github репозиторий pizza-margherita

Настройка и подключение

Terrapizza.by

Шаг 1:

Создаем репозиторий с именем pizza-margherita на своем репозитории Github.

Добавляем файл .gitignore и вписываем название папок и расширения файлов для игнорирования.

```
# Default ignored files  
/shelf/  
/workspace.xml
```

```
# IntelliJ  
.idea/  
*.iml  
*.iws
```

```
# Maven  
log/  
target/
```

```
# Package Files #  
*.jar  
*.war  
*.nar  
*.ear  
*.zip  
*.tar.gz  
*.rar
```

```
# Log file  
*.log
```

```
# Compiled class file  
*.class
```

Настройка и подключение

Terrapizza.by

Шаг 2:

Заходим в репозиторий pizza-margherita и создаем копию удаленного репозитория при помощи команды:

```
git clone <name_repository>
```

Шаг 3:

Создаем проект pizza-margherita и именуем его и подключаем библиотеки.

Библиотеки были взяты с [mvnrepository](https://mvnrepository.com).

Код:

```
<dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>5.8.2</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.1.1</version>
  </dependency>
  <dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>4.1.0</version>
  </dependency>
</dependencies>
```

Создание pageObject

Terrapizza.by

- Создаем class AbstractPage
- Добавляем URL сайта
<https://terrapizza.by/>
- Объявляем переменные
- Создаем конструктор

```
public class AbstractPage {  
    protected final static String BASE_URL = "https://terrapizza.by/";  
  
    protected WebDriver driver;  
    protected final static int WAIT_TIMEOUT_SECONDS = 5;  
  
    protected AbstractPage(WebDriver driver) {  
        this.driver = driver;  
    }  
}
```

Создание pageObject

Terrapizza.by

- Создаем class AbstractTest
- Объявляем переменную
- Создаем методы
- Добавляем аннотации к методам @BeforeEach и @AfterEach

```
public class AbstractTest {  
    protected static WebDriver driver;  
  
    @BeforeEach  
    public void setUp() {  
        WebDriverManager.chromedriver().setup();  
        driver = new ChromeDriver();  
        driver.manage().window().maximize();  
    }  
  
    @AfterEach  
    public void tearDown() {  
        driver.quit();  
    }  
}
```

Создание pageObject

Terrapizza.by

- Создаем class HomePage и наследуем его от AbstractPage. В данном классе будут реализованы методы:
 - Открыть страницу
 - Найти и нажать кнопку меню «Пицца»
 - Найти и нажать кнопку меню «Бар»
- Объявляем переменные и добавляем xPath
- Создаем конструктор и методы

```
public class HomePage extends AbstractPage {  
  
    @FindBy(xpath = "//li[@class='menu-navigation__item']/a[@href='/menu/cat/picca']")  
    private WebElement buttonMenuPizza;  
  
    @FindBy(xpath = "//li[@class='menu-navigation__item']/a[@href='/menu/cat/bar']")  
    private WebElement buttonMenuBar;  
  
    public HomePage(WebDriver driver) {  
        super(driver);  
        PageFactory.initElements(driver, this);  
    }  
  
    public HomePage openPage(){  
        driver.get(BASE_URL);  
        return this;  
    }  
  
    public HomePage clickButtonMenuPizza(){  
        buttonMenuPizza.click();  
        return this;  
    }  
  
    public HomePage clickButtonMenuBar(){  
        buttonMenuBar.click();  
        return this;  
    }  
}
```


Создание pageObject

Terrapizza.by

- Создаем class PizzaPage и наследуем его от AbstractPage. В данном классе будут реализованы методы:
 - Добавить пиццу «Маргарита» и в корзину
 - Найти и открыть корзину
 - Проверить добавление пиццы «Маргарита» в корзину
 - Закреть корзину
- Объявляем переменные и добавляем xPath
- Создаем конструктор и методы

```
public class PizzaPage extends AbstractPage{

    @FindBy(xpath = "//div[@class='cart-button']/button[@data-id='364']")
    private WebElement buttonAddPizzaToCart;
    @FindBy(xpath = "//div[@class='basket__btn-top basket__top basket__btn-top--sm']")
    private WebElement buttonOrder;
    @FindBy(xpath = "//li[@id='basket-el-0']//div[@class='basket__products-item-name']")
    private WebElement labelPizza;
    @FindBy(xpath = "//div[@id='basket-btn']")
    private WebElement buttonOrderOpened;

    public PizzaPage(WebDriver driver) {
        super(driver);
        PageFactory.initElements(driver, this);
    }

    public PizzaPage clickButtonAddPizzaToCart(){
        buttonAddPizzaToCart.click();
        return this;
    }

    public boolean isPizzaInCart(String pizzaName){
        return labelPizza.getText().contains(pizzaName);
    }
}
```

```
public PizzaPage clickButtonOrder(){
    new WebDriverWait(driver,
        Duration.ofSeconds(WAIT_TIMEOUT_SECONDS))
        .until(ExpectedConditions.elementToBeClickable(buttonOrder))
        .click();
    return this;
}

public PizzaPage clickCloseButtonOrderOpened(){
    new WebDriverWait(driver,
        Duration.ofSeconds(WAIT_TIMEOUT_SECONDS))
        .until(ExpectedConditions.elementToBeClickable(buttonOrderOpened))
        .click();
    return this;
}
}
```

Создание pageObject

Terrapizza.by

- Создаем class BarPage и наследуем его от AbstractPage. В данном классе будет реализованы методы:
 - Добавить напиток «Имбирь-клюква» в корзину
 - Найти и открыть корзину
 - Проверить добавление напитка «Имбирь-клюква» в корзину
- Объявляем переменные и добавляем xPath
- Создаем конструктор и методы

```
public class BarPage extends AbstractPage{
    @FindBy(xpath = "//div[@class='cart-button']//button[@data-id='1451']")
    private WebElement buttonAddDrinkToCart;
    @FindBy(xpath = "//div[@class='basket__btn-top basket__top basket__btn-top--sm']")
    private WebElement buttonOrder;
    @FindBy(xpath = "//li[@id='basket-el-1']//div[@class='basket__products-item-name']")
    private WebElement labelDrink;

    public BarPage(WebDriver driver) {
        super(driver);
        PageFactory.initElements(driver, this);
    }

    public BarPage clickButtonAddDrinkToCart(){
        buttonAddDrinkToCart.click();
        return this;
    }

    public boolean isDrinkInCart(String drinkName){
        return labelDrink.getText().contains(drinkName);
    }
}
```

```
public BarPage clickButtonOrder(){
    new WebDriverWait(driver,
        Duration.ofSeconds(WAIT_TIMEOUT_SECONDS))
        .until(ExpectedConditions.elementToBeClickable(buttonOrder))
        .click();
    return this;
}
```

Test

Terrapizza.by

- Создаем class TerrapizzaTest и наследуем его от AbstractTest. В данном классе будет выполнен тестовый сценарий:
 - Открыть страницу
 - Перейти в меню «Пицца»
 - Добавить пиццу «Маргарита»
 - Открыть корзину и закрыть корзину
 - Перейти в меню «Бар»
 - Добавить напиток «Имбирь-клюква» в корзину
 - Открыть корзину
 - Проверить добавление пиццы «Маргарита» и напитка «Имбирь-клюква» в корзину
- Объявляем переменные для сравнения

```
public class TerrapizzaTest extends AbstractTest {
    HomePage homePage;
    PizzaPage pizzaPage;
    BarPage barPage;

    private final String pizzaName = "Пицца Маргарита";
    private final String drinkName = "Имбирь-клюква";

    @Test
    public void testAddPizzaToCart() throws
        InterruptedException {
        homePage = new HomePage(driver)
            .openPage()
            .clickButtonMenuPizza();

        pizzaPage = new PizzaPage(driver)
            .clickButtonAddPizzaToCart()
            .clickButtonOrder()
            .clickCloseButtonOrderOpened();

        homePage
            .clickButtonMenuBar();
    }
}
```

```
barPage = new BarPage(driver)
    .clickButtonAddDrinkToCart()
    .clickButtonOrder();

assertTrue(pizzaPage.isPizzaInCart(pizzaName));
assertTrue(barPage.isDrinkInCart(drinkName));
//Thread.sleep(5000);
```

Terrapizza.by

Спасибо за внимание