

Aufgabe 4: Page Object Model (POM)

Ziel:

Sie übertragen die bisherigen Tests auf ein Page-Object-Design, um die Trennung von Seitenbeschreibung und Testlogik zu realisieren und die Wartbarkeit zu steigern.

Szenario:

Bisher haben Sie in den Tests direkt mit Selenium-WebElements gearbeitet. In dieser Aufgabe extrahieren Sie alle Locator-Definitionen und Interaktionsmethoden in eigene Page-Object-Klassen und lassen Ihre Testklassen nur noch über diese Abstraktionen agieren.

Aufgaben:

1. Web-Calculator-Tests mit POM

- Erstellen Sie die Page-Object-Klasse **CalculatorPage**, die alle Locator-Properties und Methoden (EnterFirstNumber(), EnterSecondNumber(), SelectOperation(), ClickCalculate(), GetResult()) kapselt.
- Implementieren Sie die Testklasse **CalculatorTests**, die ausschließlich diese Methoden verwendet, um Addition, Subtraktion, Multiplikation und Division zu validieren.

2. Produktsuche-Tests mit POM

- Page-Object:
 - Erstellen Sie SearchPage für <https://automationteststore.com> mit Methoden:
 - Definieren Sie dazu die Locator-Properties (Suchfeld, Suchbutton, erstes Produkt-Title-Locator, Ergebnis-Items-Locator, „keine Treffer“-Locator)
- Testklasse:
 - Implementieren Sie SearchTests mit drei Szenarien:
 1. ProduktSucheEinProdukt
 - Suche nach "Flip Flop", vergleiche den ersten Produkt-Titel mit "New Ladies High Wedge Heel Toe Thong Diamante Flip Flop Sandals".
 2. ProduktSucheMehrereProdukte
 - Suche nach "T-Shirt", prüfe die Ergebnisse
 3. ProduktSucheKeinProdukt
 - Suche nach "xyz123nomatch", vergleiche mit "There is no product that matches the search criteria."