Page Object Model

Why Page Object Model?

- Building a test framework
- Easily maintain and update page elements
- Create simple, clean tests
- Simplify the purpose for each class

So far, we have written our tests in one class file.

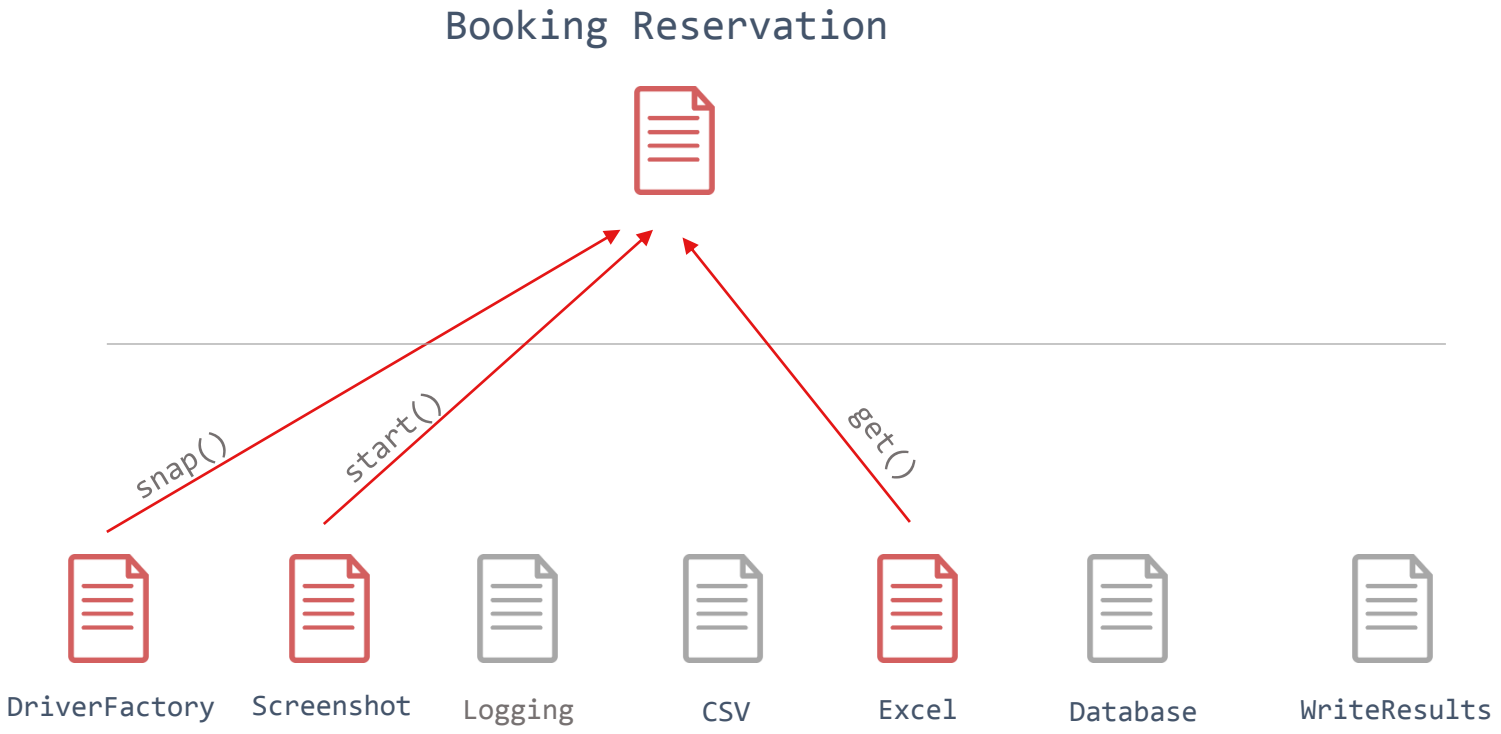
Challenges with Tests:

- *Not very readable*
- *Not easy to update page elements*
- *Not easy to modify test steps*
- *What if we wanted to run a test involving many of the same steps?*

Framework

tests

utilities


Booking Reservation
snap()
start()
get()
DriverFactory
Screenshot
Logging
CSV
Excel
Database
WriteResults

Page Object Model Framework

`ShoppingCart`

Home Page
Product Search
Add to Cart
Modify Cart
Checkout
Confirmation

`CheckCart`

Home Page
User Login
View Cart
Modify Cart

Page Object Model Framework

`ShoppingCart`

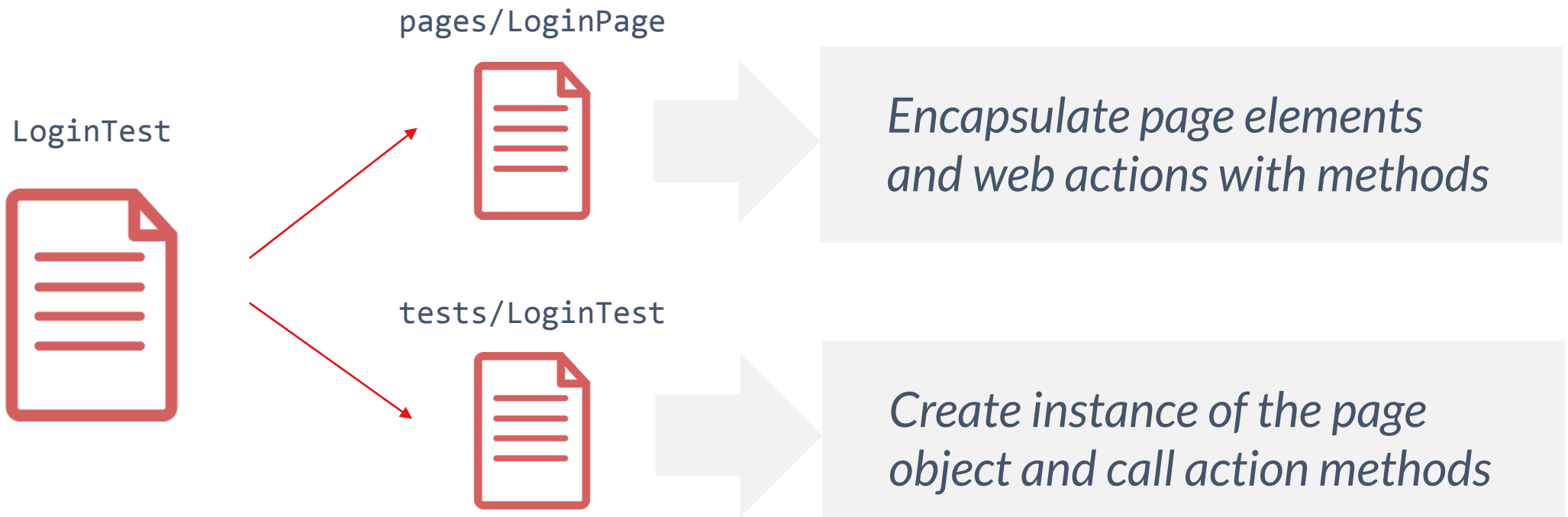
Home Page
Product Search
Add to Cart
Modify Cart
Checkout
Confirmation

`CheckCart`

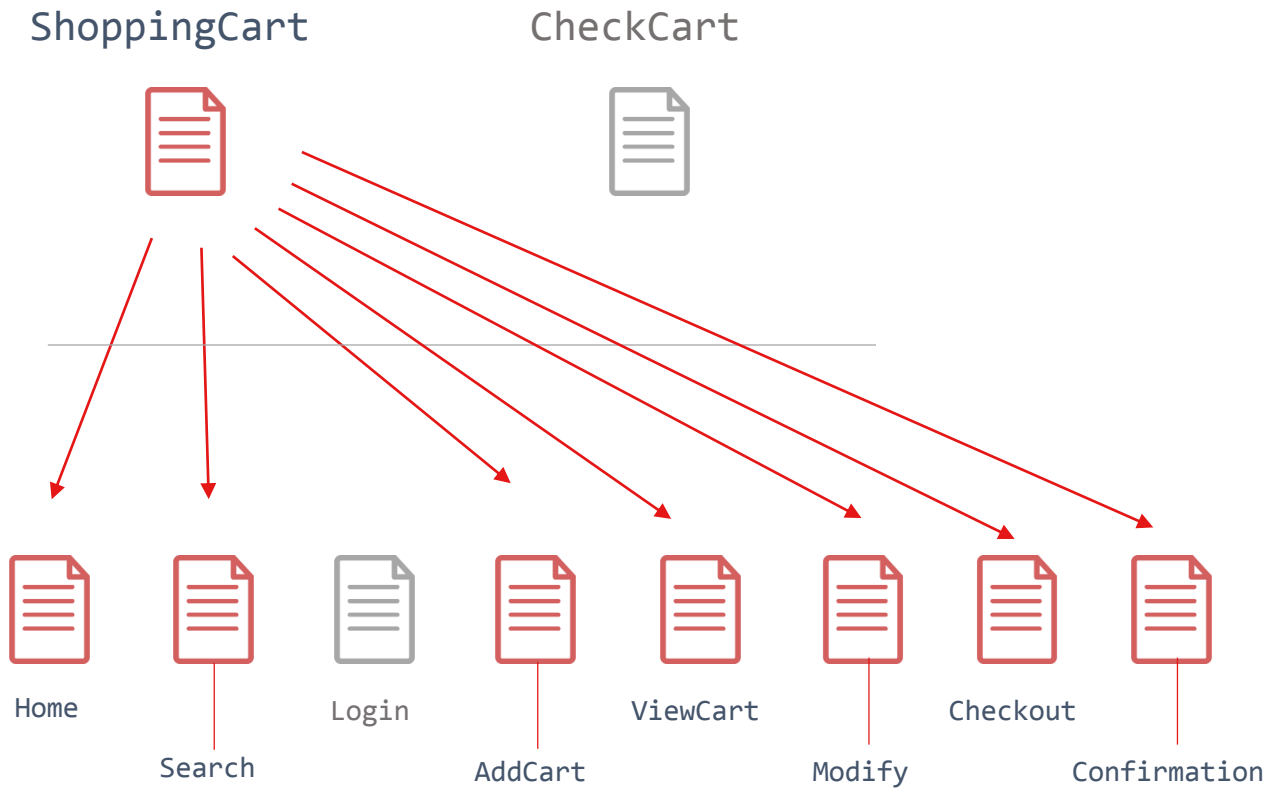
Home Page
User Login
View Cart
Modify Cart

POM Solution: Split into Two Classes

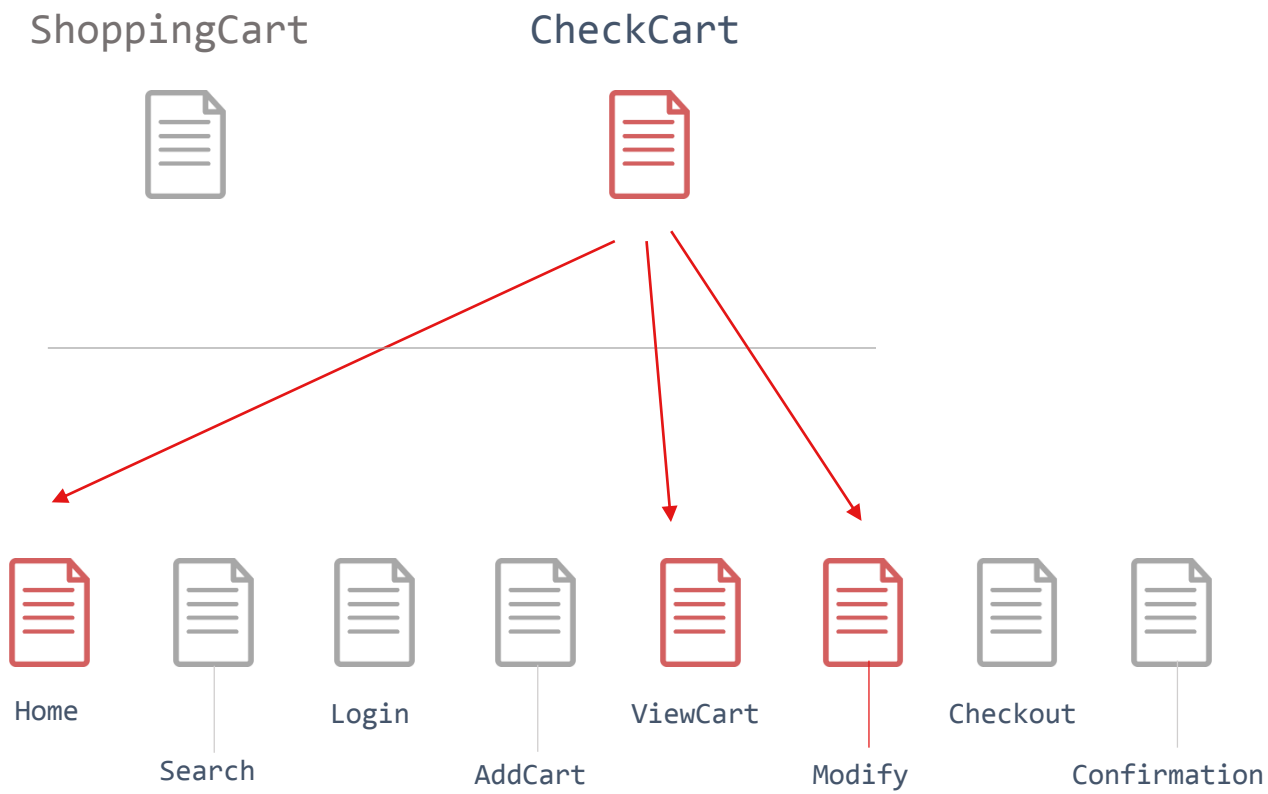
One for the test steps, one for the page elements



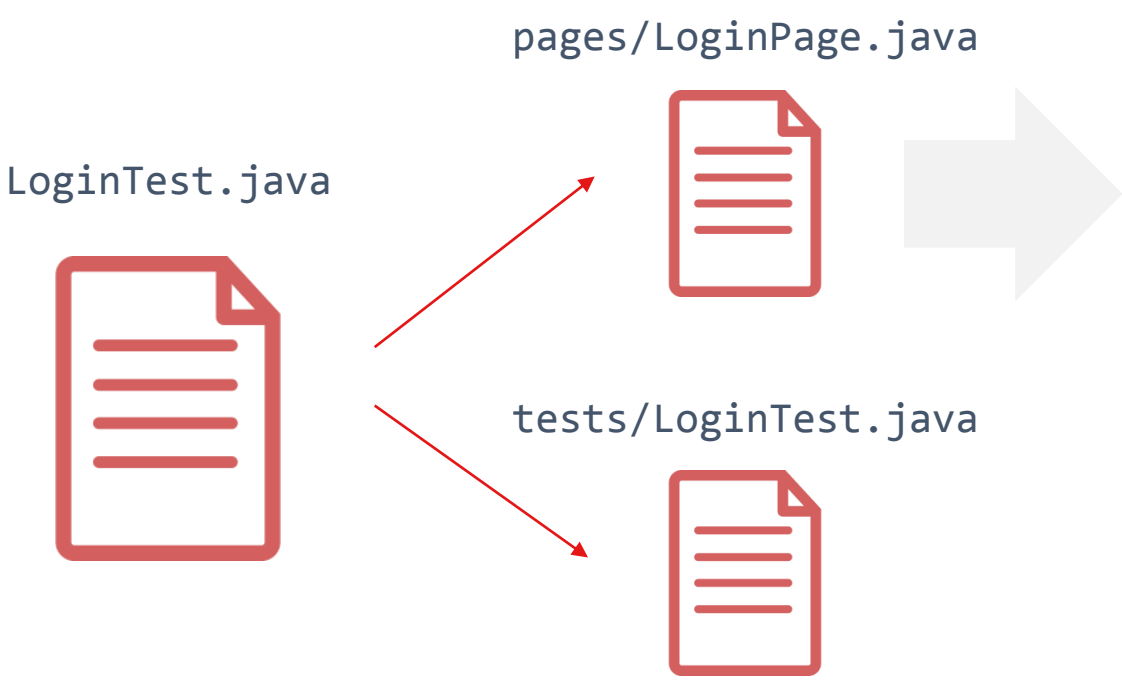
Page Object Model Framework



Page Object Model Framework



Enhance POM with PageFactory



Use the constructor method to initialize Web Elements using the PageFactory.init method