**Modular & Granular:**

* Modularity concerns breaking up systems into separate parts.
* Granularity deals with the size of those separate parts.

**What is a modular approach?**

A modular approach involves dividing a complex system into independent, manageable units. Each module functions separately yet can be integrated as a whole to achieve a goal/task

**What is a granular approach?**

How thoroughly a system or its observation is dissected into smaller parts

**What is the Page Object Model (POM)? - Modularise our code**

Page Object Model is a design pattern commonly used in test automation for creating an Object Repository of web elements. The primary advantage of adopting POM is to reduce code duplicity and to reduce maintenance efforts.

*Page Object Model is a design pattern used in Automation, where we create an object repository to store web elements*. *A java class is created that corresponds to each web page, consisting of the webElements on the page and the corresponding methods that act on elements.* Multiple test scripts can further use these reusable methods to perform various web actions. Since a separate class file is created for each web page and the common locators can be used by multiple test classes, this reduces the code duplicity and improves code maintenance.

**Advantages of Page Object Model:**

***Object Repository****: we can use the same object repository for a different purpose with different tools. For example, we can integrate Page Object Model in Selenium with TestNG/JUnit for functional*Testing *and at the same time with Cucumber for acceptance testing.*

***Reusability****: We can reuse the page class if required in different test cases.*

***Functional encapsulation****: All possible operations that can be performed on a page can be defined and contained within the same class created for each page. This allows for clear definition and scope of each page functionality.*

***Low maintenance****: Any user interface changes can swiftly be implemented into the interface as well as class.*

***Low Redundancy****: Helps to reduce duplication of data.*

***Programmer Friendly****: Robust and more readable. The Object Oriented approach makes framework programmer friendly.*

**What is Page Factory in Selenium?**

*Page Factory is a class provided by Selenium Web Driver to support Page Object Design patterns. In Page Factory, testers use @FindBy annotation. The initElements method is used to initialize web elements.* Page Factory is an extension of the Page Object Model pattern, and it uses annotations to locate and initialize the elements on the page. This reduces the amount of code needed to locate the elements and makes the code more readable and maintainable.

***@FindBy****can accept****tagName, partialLinkText, name, linkText, id, css, className, xpath****as attributes.*

**Advantages of Page Factory:**

1. ***Easy Initialization of Web Elements****: With PageFactory, you can use annotations like @FindBy to declare web elements directly within the page object class. These annotations are used to locate elements on the web page using various locators (e.g., id, name, xpath, etc.). PageFactory automatically initializes the web elements when the page object is instantiated, eliminating the need for explicit element initialization.*

*2.****Lazy Initialization****: PageFactory provides lazy initialization of web elements, meaning the elements are initialized only when they are accessed or interacted with in the test code. This approach optimizes the performance by avoiding unnecessary element lookup and initialization when the elements are not needed.*

3.***Improved code readability****: Page Factory makes the automation code more readable by separating the initialization of web elements from the test code. This makes it easier to understand the intent of the code and the interactions with the web page.*

4. ***Better test performance****: Page Factory can improve test performance by reducing the overhead of locating web elements on a web page. This is achieved by initializing the Page Object only once per test, rather than for each test method.*

A screenshot of a web page

Description automatically generated

| **POM** | **PageFactory** |
| --- | --- |
| It is an approach for Design patterns. | It is a class that is provided by Selenium Webdriver. |
| It helps in separating page objects and test scripts. | It is a technique to implement POM. |
| ‘By’ annotation is used to define page objects. | It uses annotation ‘FindBy’ to describe page objects. |
| It requires the initialization of every object. | It does not require the initialization of every object. |
| There is cache storage for performing tasks. | There is no need for cache storage. |

* How to get the driver to the page object class – ans: through constructor(parameterised)
* How does @FindBy uses the driver to find elements – ans: through the PageFactory.initElements(driver, this) method declared in constructor
* Page Object should only hold elements and actions – No data should be residing here!
* Also all assertions should happen from the test case and not from page objects