

Use of OPPS concepts in Selenium

OOPs concept in Selenium Framework

1. **INTERFACE:** `WebDriver driver = new ChromeDriver();`
In this statement `WebDriver` is nothing but interface in selenium.
2. **UPCASTING:** `WebDriver driver = new ChromeDriver();`
above statement is nothing but UPCASTING in selenium.
3. **INHERITANCE:** We create a Base Class in the Framework to initialize `WebDriver` interface, `WebDriver` waits, Property files etc., in the Base Class. We extend the Base Class in Tests Class. that is nothing but Inheritance in Selenium Framework.
4. **POLYMORPHISM:** Combination of overloading and overriding is known as Polymorphism.
5. **METHOD OVERLOADING:** We use implicit wait in Selenium. Implicit wait is an example of overloading. In Implicit wait we use different time stamps such as SECONDS, MINUTES, HOURS etc., A class having multiple methods with same name but different parameters is called Method Overloading.

eg. `driver.switchTo().frame()`: - String name, int index
6. **METHOD OVERRIDING:** Declaring a method in child class which is already present in the parent class is called Method Overriding.

Examples are `get` and `navigate` methods of different drivers in Selenium.
7. **ENCAPSULATION:** All the POM classes in a framework are an example of Encapsulation. In POM classes, we declare the data members using `@FindBy` and initialization of data members will be done using Constructor to utilize those in methods. Encapsulation is a mechanism of binding code and data together in a single unit. Encapsulation is the process of wrapping up code and data together in a single unit. It is used to hide the data of a class from another class. Encapsulation can be achieved when you declare all variables as private and a public method in a class to get the values of the variable.
8. **ABSTRACTION:** In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) in a Page Class. We utilize these locators in pom class but we can't see these locators in the tests. Literally we hide the locators from the tests. Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.