15 Tricky Java + Selenium Interview Programs

Handling StaleElementReferenceException

Occurs when DOM refreshes after locating element. Retry logic helps handle it gracefully.

```
public WebElement retryingFind(By locator) {
    WebElement element = null;
    int attempts = 0;
    while (attempts < 3) {
        try {
            element = driver.findElement(locator);
            break;
        } catch (StaleElementReferenceException e) {
            System.out.println("Retrying... attempt " + attempts);
        }
        attempts++;
    }
    return element;
}
```

FluentWait Instead of Thread.sleep

Avoid Thread.sleep() using FluentWait with polling interval.

```
Wait<WebDriver> wait = new FluentWait<>(driver)
    .withTimeout(Duration.ofSeconds(30))
    .pollingEvery(Duration.ofSeconds(2))
    .ignoring(NoSuchElementException.class);
WebElement element = wait.until(d -> d.findElement(By.id("submit")));
```

Dynamic Dropdown Using Streams

Use Java Streams to handle dynamic dropdowns elegantly.

```
List<WebElement> options = driver.findElements(By.xpath("//ul/li"));
options.stream()
    .filter(e -> e.getText().equalsIgnoreCase("India"))
    .findFirst()
    .ifPresent(WebElement::click);
```

Highlight Element Before Click

Use JavaScriptExecutor to visually highlight an element before performing an action.

```
WebElement ele = driver.findElement(By.id("login"));
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("arguments[0].style.border='3px solid #50FA7B'", ele);
```

Verify Sorting in Table

Compare sorted data from UI with programmatic sorting.

```
List<String> names = driver.findElements(By.xpath("//table//td[1]"))
    .stream().map(WebElement::getText).collect(Collectors.toList());
List<String> sorted = new ArrayList<>(names);
Collections.sort(sorted);
Assert.assertEquals(names, sorted);
```

Capture Broken Links

Validate hyperlinks using HTTP response codes.

```
List<WebElement> links = driver.findElements(By.tagName("a"));
for (WebElement link : links) {
    String url = link.getAttribute("href");
    if (url == null || url.isEmpty()) continue;
    HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
    conn.setRequestMethod("HEAD");
    conn.connect();
    int code = conn.getResponseCode();
    if (code >= 400) System.out.println(url + " is broken: " + code);
}
```

File Upload (AutoIt or Robot)

Handle native dialogs using AutoIt or Robot when sendKeys() isn’t possible.

```
// Using AutoIt
Runtime.getRuntime().exec("C:\\path\\to\\AutoIt\\UploadFile.exe");

// Or using Robot
Robot robot = new Robot();
StringSelection ss = new StringSelection("C:\\path\\to\\file.txt");
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(ss, null);
robot.keyPress(KeyEvent.VK_CONTROL);
```

```
robot.keyPress(KeyEvent.VK_V);
robot.keyRelease(KeyEvent.VK_CONTROL);
robot.keyRelease(KeyEvent.VK_V);
robot.keyPress(KeyEvent.VK_ENTER);
robot.keyRelease(KeyEvent.VK_ENTER);
```

File Download Verification

Ensure the downloaded file exists within a timeout window.

```
File downloaded = new File(System.getProperty("user.home") + "/Downloads/file.txt");
new WebDriverWait(driver, Duration.ofSeconds(30))
    .until(d -> downloaded.exists());
Assert.assertTrue(downloaded.exists());
```

Window Handling (Check Title, Then Switch)

Switch between windows based on title verification.

```
String parent = driver.getWindowHandle();
for (String handle : driver.getWindowHandles()) {
    driver.switchTo().window(handle);
    if (driver.getTitle().contains("Expected Title")) break;
}
driver.switchTo().window(parent);
```

Screenshot with Failed Test Name

Capture screenshots with test name in the filename for clarity.

```
String testName = result.getMethod().getMethodName();
File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
String ts = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
FileUtils.copyFile(src, new File("./screenshots/" + testName + "_" + ts + ".png"));
```

Retry Failed Test Cases (TestNG)

Use IRetryAnalyzer to rerun failed test cases automatically.

```
public class RetryAnalyzer implements IRetryAnalyzer {
    int count = 0, maxTry = 2;
    public boolean retry(ITestResult result) {
        if (count < maxTry) { count++; return true; }
        return false;
    }
}
```

Scroll Until Element Visible

Scroll until element is visible using JavaScriptExecutor.

```
WebElement ele = driver.findElement(By.id("footer"));
((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);",
ele);
```

Wait Until Page Load Completes

Wait for full DOM readiness using document.readyState.

```
new WebDriverWait(driver, Duration.ofSeconds(30))
    .until(d -> ((JavascriptExecutor) d)
    .executeScript("return document.readyState").equals("complete"));
```

Capture Browser Console Errors

Fetch and log JavaScript console errors.

```
LogEntries logs = driver.manage().logs().get(LogType.BROWSER);
for (LogEntry entry : logs) {
    System.out.println("Console: " + entry.getMessage());
}
```

Read JSON Using ObjectMapper

Use Jackson ObjectMapper for easy JSON parsing.

```
ObjectMapper mapper = new ObjectMapper();
Map<String, Object> data = mapper.readValue(new File("data.json"), new
TypeReference<Map<String,Object>>(){});
String username = (String) data.get("username");
```