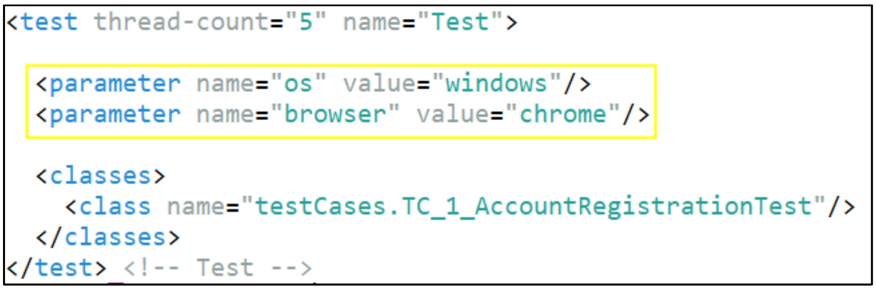
Creating an Automated TestNG XML File

- To create a TestNG XML file, right-click on the test case class `TC_1_AccountRegistrationTest` → select **“Convert to TestNG”**.
- In the generated `master.xml` file, pass two parameters — **Browser Name** and **Operating System** — as shown below:

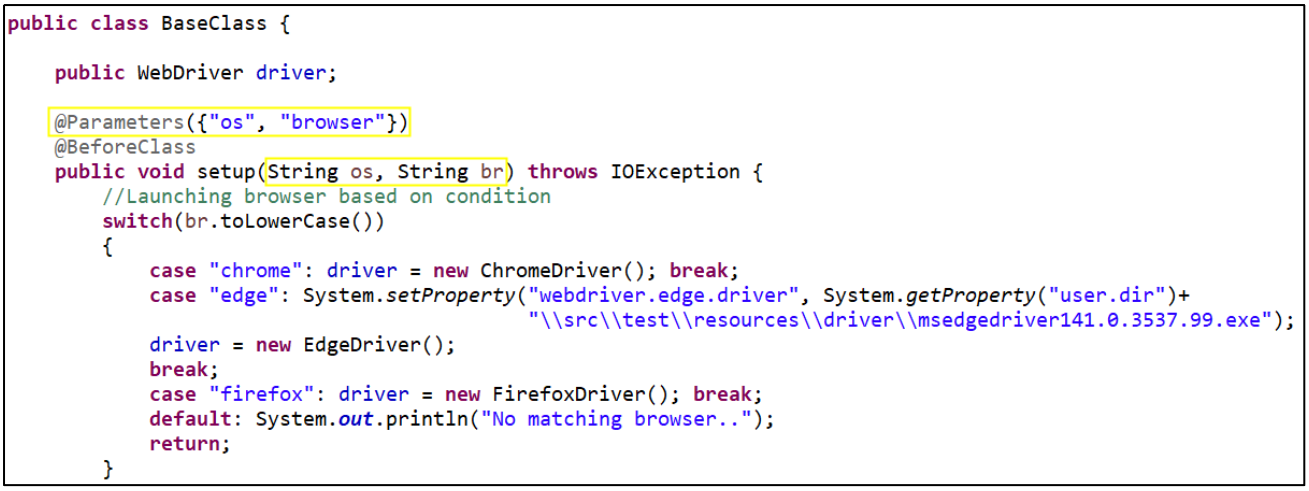


```
<test thread-count="5" name="Test">

  <parameter name="os" value="windows"/>
  <parameter name="browser" value="chrome"/>

  <classes>
    <class name="testCases.TC_1_AccountRegistrationTest"/>
  </classes>
</test> <!-- Test -->
```

- Receive both parameters in the `setup()` method of the `BaseClass`, as this method is responsible for launching the browser.
- Use the `@Parameters` annotation in the `BaseClass` to read these values dynamically, enabling cross-browser and cross-platform test execution.



```
public class BaseClass {

    public WebDriver driver;

    @Parameters({"os", "browser"})
    @BeforeClass
    public void setup(String os, String br) throws IOException {
        //Launching browser based on condition
        switch(br.toLowerCase())
        {
            case "chrome": driver = new ChromeDriver(); break;
            case "edge": System.setProperty("webdriver.edge.driver", System.getProperty("user.dir")+
                                            "\\src\\test\\resources\\driver\\msedgedriver141.0.3537.99.exe");
            driver = new EdgeDriver();
            break;
            case "firefox": driver = new FirefoxDriver(); break;
            default: System.out.println("No matching browser..");
            return;
        }
```

Creating a TestNG XML File for Parallel Testing

- Create another TestNG XML file named `crossbrowsertesting.xml` for executing tests in **parallel** across multiple browsers.
- The `crossbrowsertesting.xml` file is used to perform a single round of **parallel testing** across multiple browsers (e.g., Chrome, Edge, Firefox), ensuring cross-browser compatibility and application stability.
- **Parallel testing** is typically executed **after completing a full validation round** on a specific browser.

Configuring Parallel Testing in TestNG

➢ Add **multiple `<test>` entries** in the TestNG XML file to achieve **parallel testing** and assign a **unique name** to each test block.

- Specify the parameter `thread-count="5"` at the **suite level** and remove it from the individual test levels.
- Add the parameter **`parallel="tests"`** at the **suite level** to enable the execution of multiple test sets in parallel.

Reading data from the properties file

The main advantage of creating a **properties file** is to manage common configuration variables (such as the application URL, login credentials, or customer IDs) so their values can be **reused across multiple test cases**, avoiding hardcoded values.