Java Access Modifiers

In Java, **access modifiers** are keywords used to define the **visibility or accessibility** of classes, methods, constructors, and variables. They determine where a member (field, method, or class) can be accessed from within your project.

🔐 Types of Access Modifiers in Java

Modifier	Class	Package	Subclass (different pkg)	World (everywhere)
`private`	✅	❌	❌	❌
default	✅	✅	❌	❌
`protected`	✅	✅	✅	❌
`public`	✅	✅	✅	✅

1. `private`

- **Scope:** Within the **same class** only.
- **Use Case:** When you want to hide internal variables/methods from outside access.
- **Example:**

```
java

class Test {
    private int data = 100;

    private void show() {
        System.out.println("Private method");
    }
}
```

2. *Default* (No modifier)

- **Scope:** Accessible **within the same package** only.
- Also called **package-private**.
- **Example:**

```
java

class Example {
    int number = 10; // default access

    void display() {
        System.out.println("Default method");
    }
}
```

3. `protected`

- **Scope:**
  - Within the same package.
  - Subclasses in other packages.
- **Use Case:** Useful in inheritance where you want to allow limited external access.
- **Example:**

```
java

class Base {
    protected void greet() {
        System.out.println("Hello from Base");
    }
}
```

---

4. `public`

- **Scope:** Accessible **from anywhere** (any class, any package).
- **Use Case:** When you want the class or method to be used freely.
- **Example:**

```
java

public class Person {
    public String name;

    public void speak() {
        System.out.println("Speaking...");
    }
}
```

📌 Access Modifiers on Classes

Only `public` and **default** are allowed for **top-level classes**:

- `public class MyClass` – accessible from anywhere.
- `class MyClass` – accessible only within the same package.

🔁 Summary Table

Modifier	Class-Level	Field	Method	Constructor
`private`	❌	✅	✅	✅
default	✅	✅	✅	✅
`protected`	❌	✅	✅	✅
`public`	✅	✅	✅	✅

Would you like a real project example showing how each modifier works in Selenium or automation testing code?

Great! Let's walk through a **realistic Java Selenium project example** that demonstrates the use of **all access modifiers** in a **Page Object Model (POM)** framework.

🔧 Project Context:

Imagine we're testing a login page using **Selenium WebDriver**, with the following:

- **`BasePage.java`** – contains reusable WebDriver code.
- **`LoginPage.java`** – POM class for login functionality.
- **`LoginTest.java`** – test class.

✅ Example Showing Access Modifiers

🗂️ `BasePage.java`

(Using `protected`, `private`, `public`)

```
java

package pages;

import org.openqa.selenium.WebDriver;

public class BasePage {

    protected WebDriver driver; // Accessible by subclasses like LoginPage

    public BasePage(WebDriver driver) {
        this.driver = driver;
    }

    private void log(String message) { // Private method used internally
        System.out.println("LOG: " + message);
    }

    public void openUrl(String url) {
        driver.get(url);
        log("Navigated to: " + url);
    }
}
```

🗂️ `LoginPage.java`

(Using `private`, `public`, and `default` access)

```
java

package pages;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;

public class LoginPage extends BasePage {

    // Default access: Only visible in this package
    By usernameField = By.id("username");
    By passwordField = By.id("password");
```

```
    private By loginBtn = By.id("login"); // Private: used only in this class

    public LoginPage(WebDriver driver) {
        super(driver);
    }

    public void login(String username, String password) {
        driver.findElement(usernameField).sendKeys(username);
        driver.findElement(passwordField).sendKeys(password);
        driver.findElement(loginBtn).click();
    }
}
```

📁 **LoginTest.java**

(Using `public` method from another package)

```
java

package tests;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import pages.LoginPage;

public class LoginTest {

    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        LoginPage loginPage = new LoginPage(driver);
        loginPage.openUrl("https://example.com/login"); // public method
        loginPage.login("admin", "admin123");           // public method

        driver.quit();
    }
}
```

🔍 How Each Access Modifier Was Used

Modifier	Where Used	Purpose
`private`	**`log()`** and **`loginBtn`**	Used only inside the class where declared
`default`	**`usernameField`**	Accessible within the same package (**`pages`**)
`protected`	**`WebDriver driver`**	Used by **`LoginPage`** which extends **`BasePage`**
`public`	**`openUrl()`**, **`login()`**	Used by the test class in a different package

---

Would you like this example as an Eclipse/IntelliJ-compatible project zip or want to add more classes like **`HomePage`**, **`ProfilePage`**, etc.?