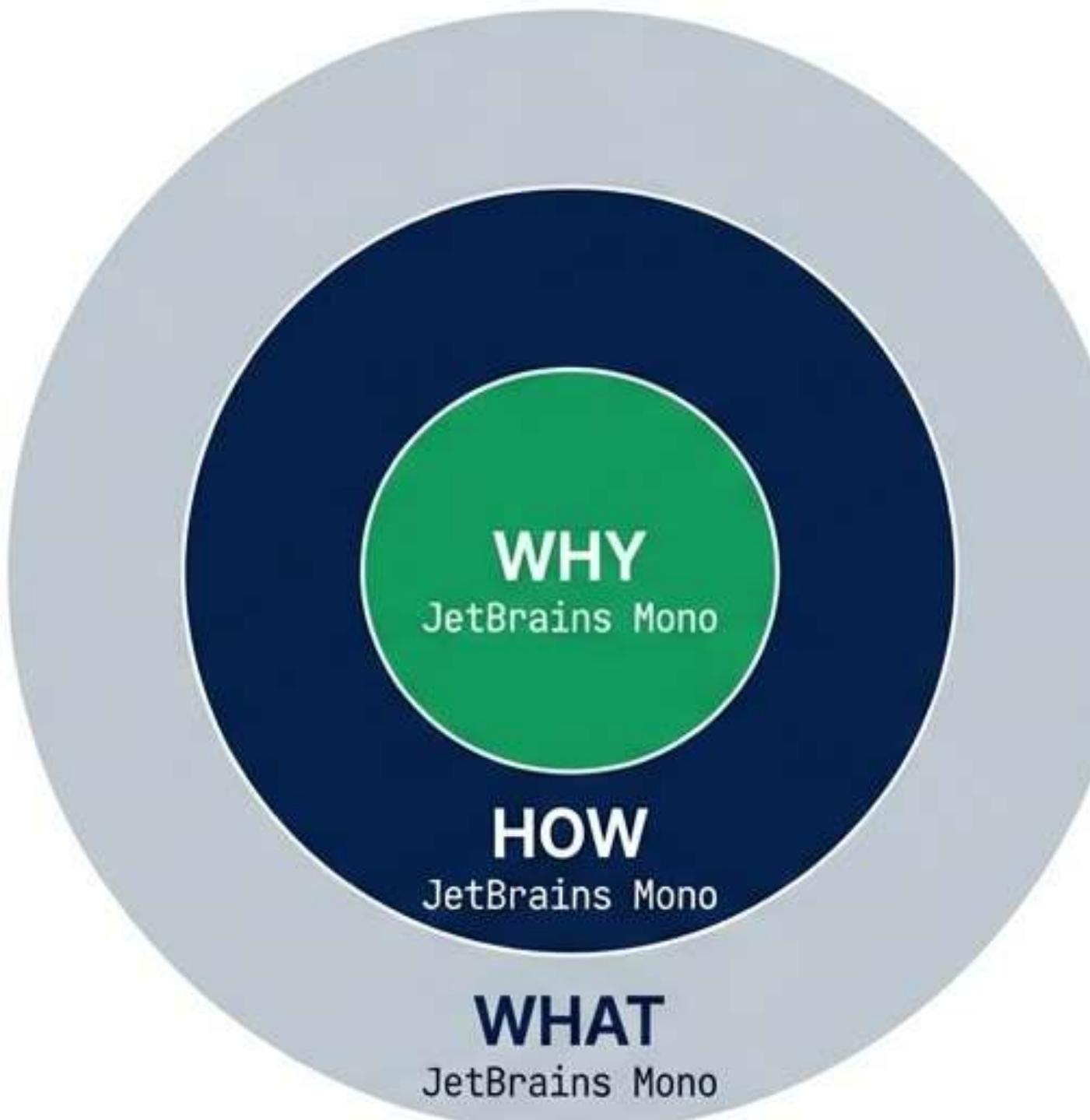




Essentials of Test Automation & Java

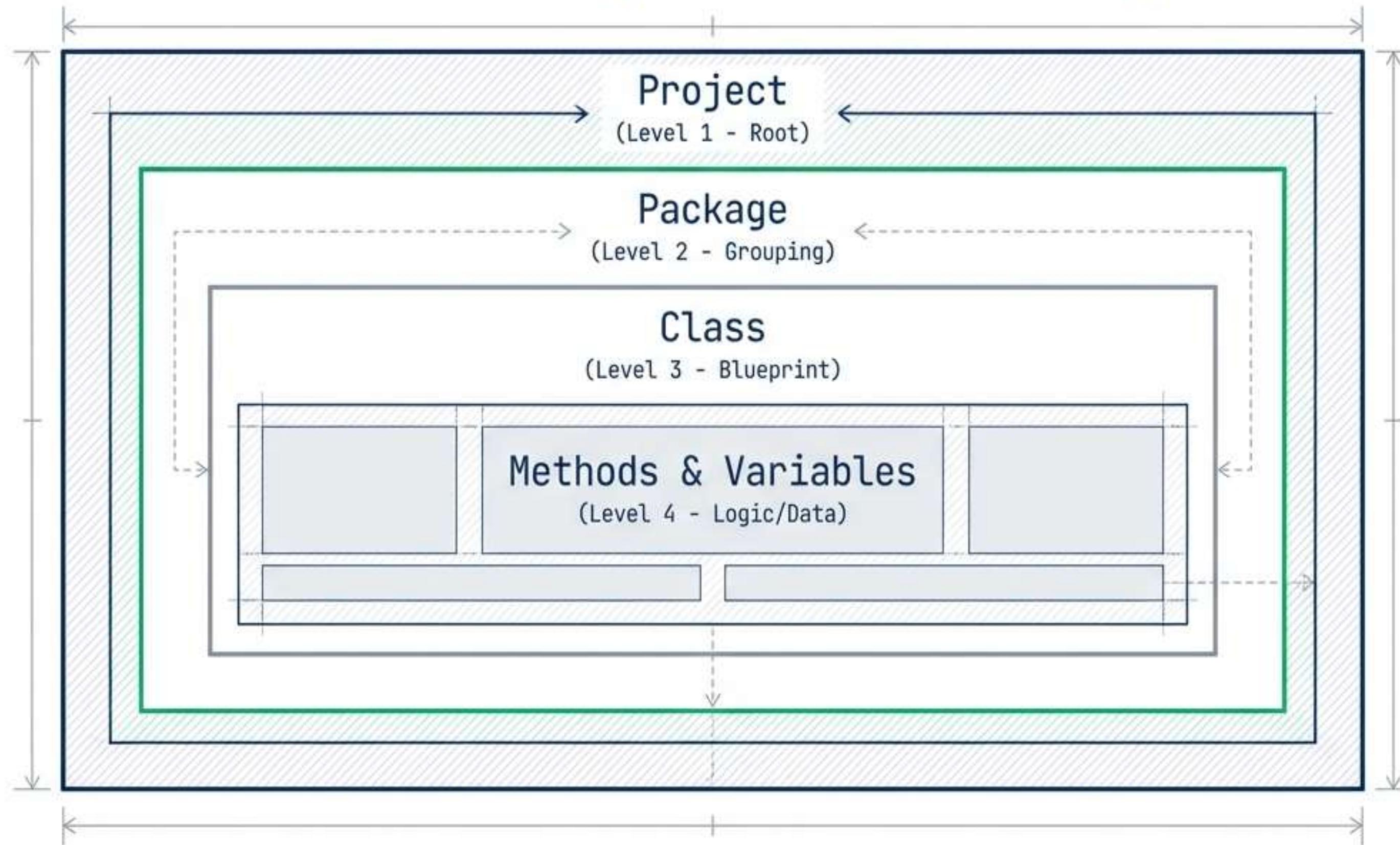
Foundation for Selenium WebDriver

The Golden Circle of Automation



- **Why:** Save time, save cost, improve quality.
- **How:** Utilising software tools (market-standard automation frameworks).
- **What:** Automating repetitive test scenarios with static logic.

The Anatomy of a Java Project



Execution Essentials

```
public static void main(String args[])
```

Entry Point: The JVM requires this specific signature to execute.

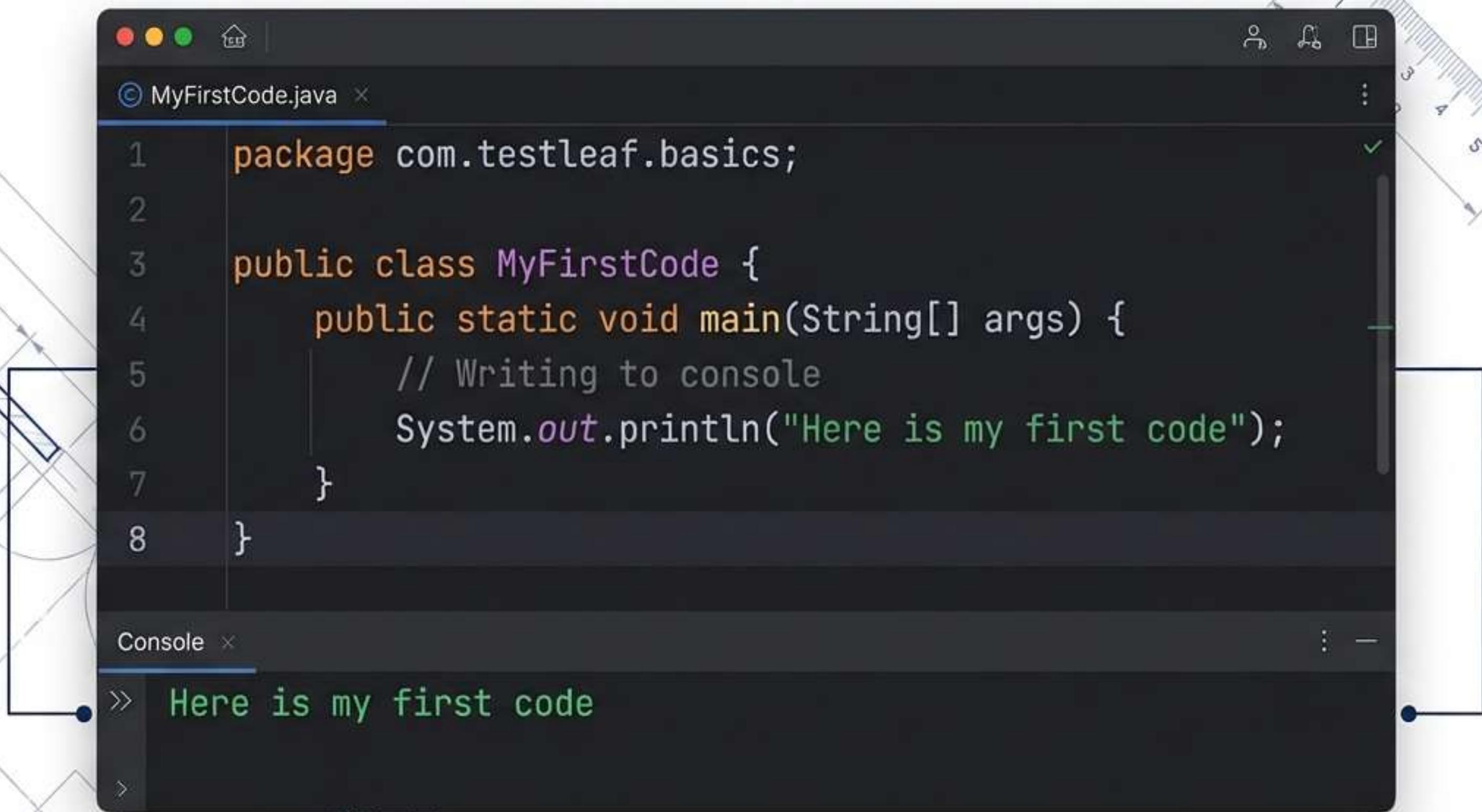
```
System.out.println()
```

Output: Prints values to the console for verification.

Pro-Tip

Use Ctrl + Space
for auto-completion.

Real-World Example



The screenshot shows a Java development environment with a code editor and a terminal window.

Code Editor: The file `MyFirstCode.java` contains the following code:

```
package com.testleaf.basics;

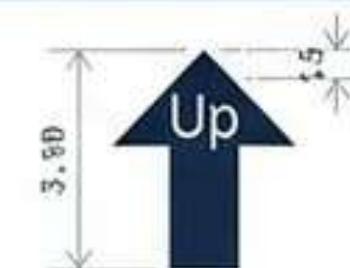
public class MyFirstCode {
    public static void main(String[] args) {
        // Writing to console
        System.out.println("Here is my first code");
    }
}
```

Terminal Window: The `Console` tab shows the output:

```
>> Here is my first code
```

Naming Conventions & Best Practices

Case Sensitivity Rule: Java distinguishes between “String” and “string”.



Upper Case (PascalCase)

- Project Names
 - MyFirstCode (from [Image-1](#))
- Class Names
 - MyFirstCode (from [Image-1](#))



Lower Case (camelCase)

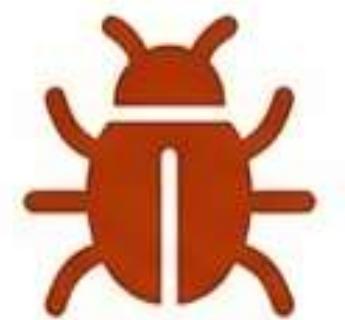
- Package Names
 - com.testleaf.basics
- Methods & Variables
 - main (from [Image-1](#))
 - args (from [Image-1](#))

Advanced Challenges: Implementation Pitfalls



Signature Mismatch

The `main` method must exactly match '**public static void main**'. Missing '**static**' or '**void**' prevents execution.



Case Sensitivity

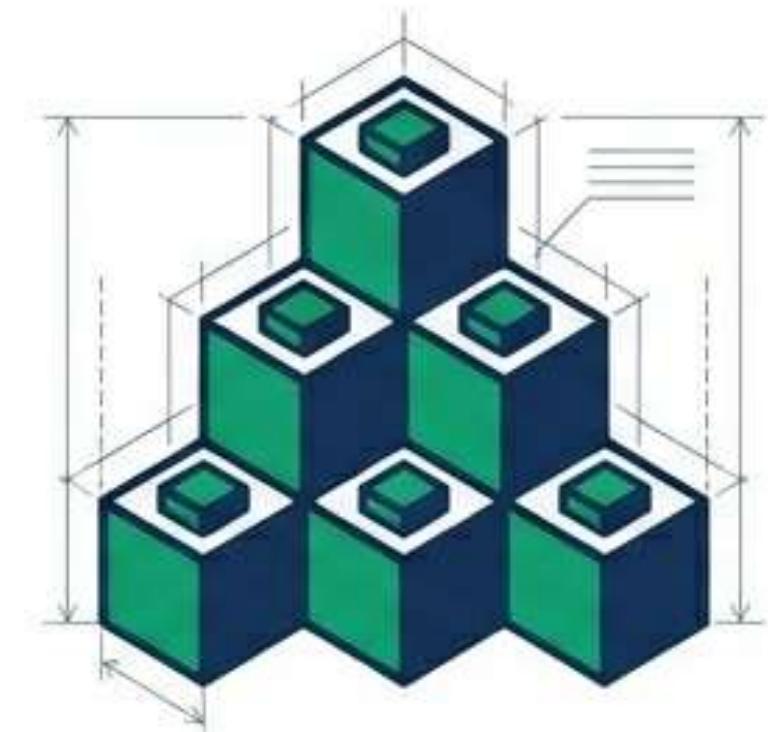
Typing 'System' as '**system**' or 'String' '**string**' will cause immediate compilation failure.



Scope Errors

Placing executable code **outside** of the Class brackets **{ }** is ← frequent logical error.

Key Takeaways

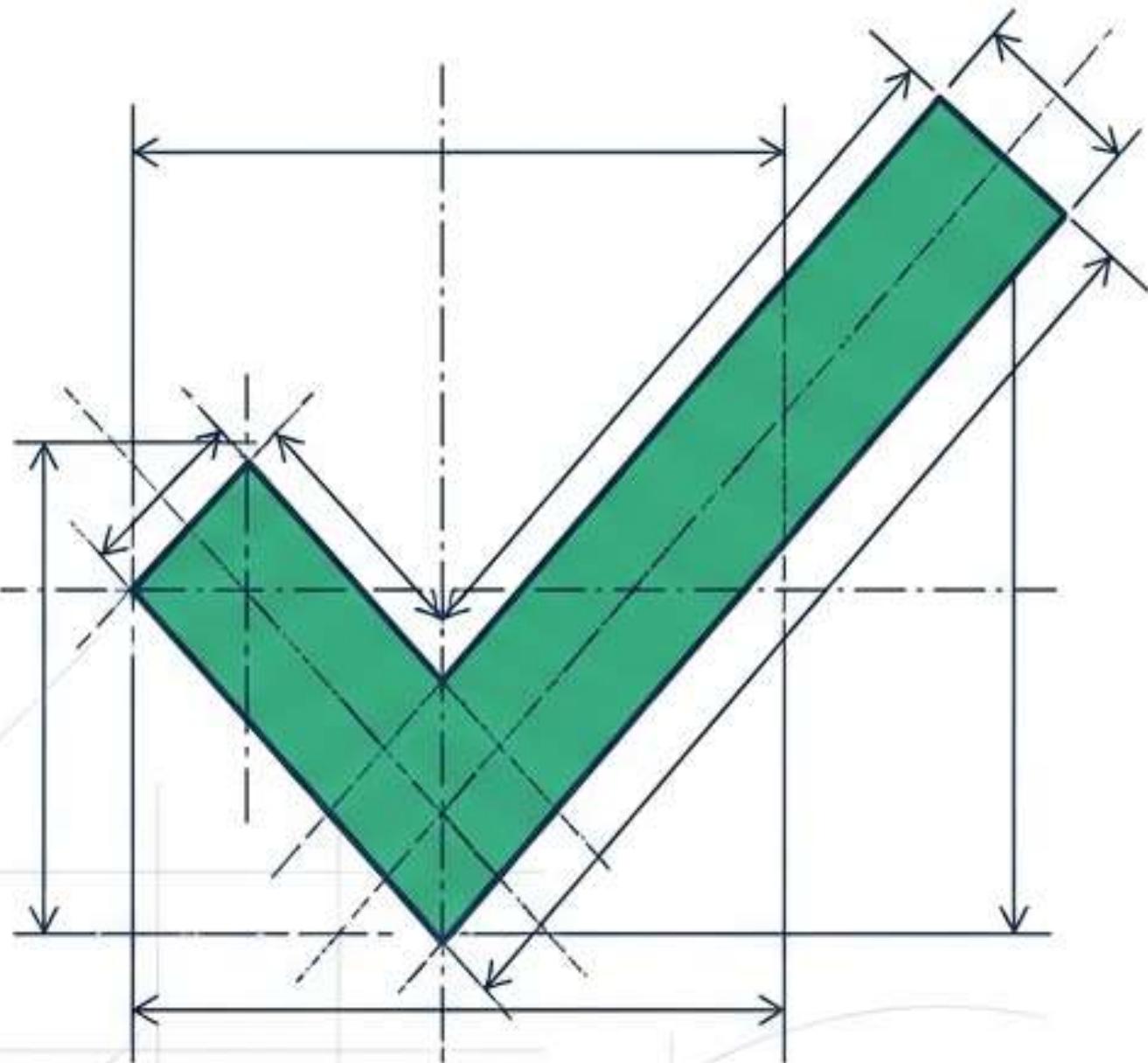


The Goal:
Automation exists to reduce repetition and cost while boosting quality.

The Structure:
Adhere to the strict hierarchy: Project > Package > Class.

The Rules: Follow Naming Conventions and ensure the main method is present for execution.

Summary



- Covered the 'Why, How, What' of Test Automation.
- Reviewed Core Java Internals (JDK/JRE) and Editor options.
- Defined Naming Conventions for Projects, Packages, and Classes.
- Executed the first script using Sysout and the main method.