The PrimeCheckApp Case Study: Test Driver Implementation & Performance

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Understand the implementation of a driver program that runs the PrimeCheck Client & measures the performance of its various methods

See github.com/douglascraigschmidt/LiveLessons/tree/master/WebMVC/ex1
Implementing the PrimeCheckTest Driver
Implementing the `PrimeCheckTest` Driver

```java
package server;

import client.PrimeCheckClient;
import common.Components;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.context.ApplicationContext;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.web.WebEnvironment;
import org.springframework.web.context.WebApplicationContext;
import java.util.List;

/**
 * This program tests the `PrimeCheckClient` and its ability to
 * communicate with the `PrimeCheckController` via Spring WebMVC
 * features.
 *
 * The `@SpringBootTest` annotation tells Spring to look for a
 * main configuration class (e.g., a `@SpringBootTest`) and use that to start a Spring application context.
 */
```
End of the PrimeCheck App Case Study: Test Driver