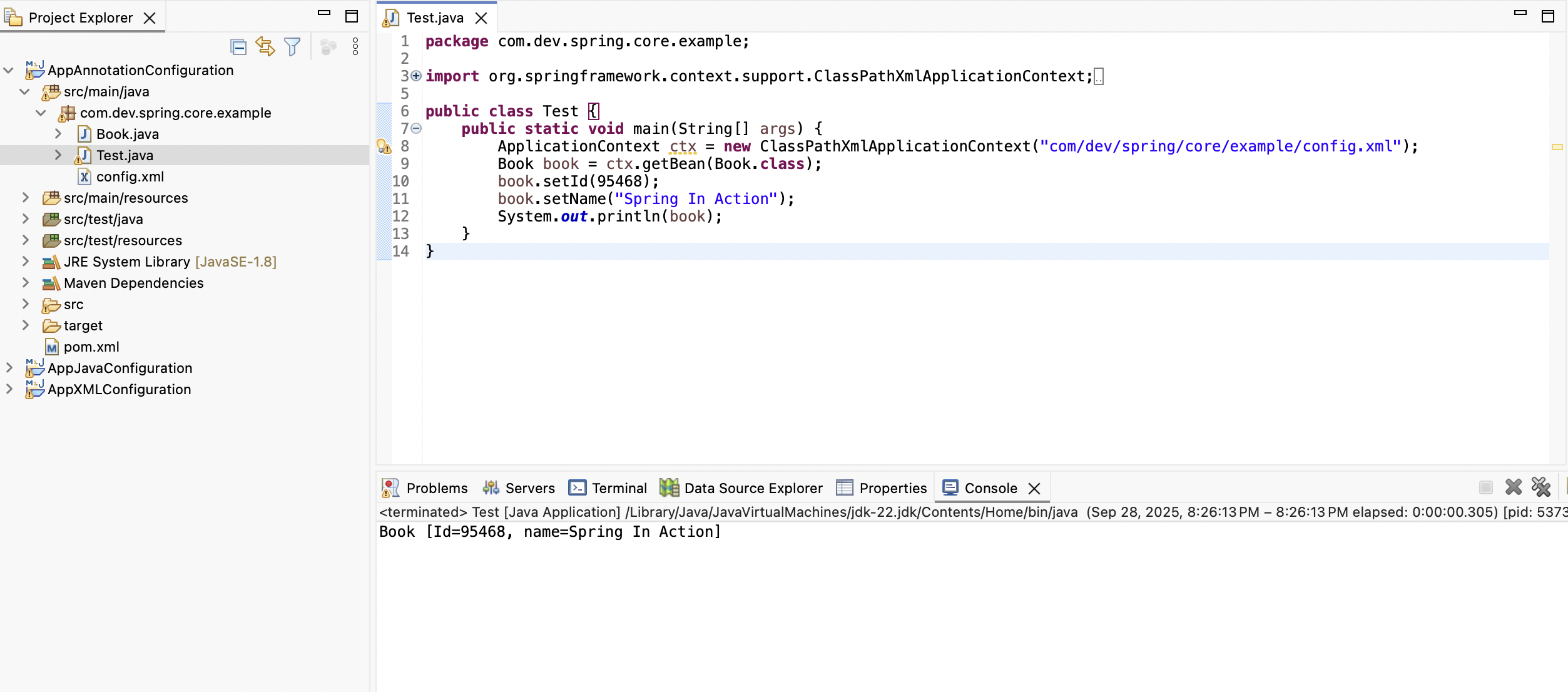
Dương Chí Thiện

Lab3:   
github:

**Part 1:**

── AppXMLConfiguration/



── AppAnnotationConfiguration/

A screen shot of a computer

AI-generated content may be incorrect.

── AppJavaConfiguration/

A screen shot of a computer

AI-generated content may be incorrect.

**Part 2:**   
Configuring XML

The conventional method in Spring, known as XML configuration, involves manually declaring beans in an external XML file. The <bean> element is used by developers to specify objects, properties, and their dependencies. Because configuration and source code are strictly kept apart using this method, changing values is simple and doesn't need recompiling the program. But as the project gets bigger, it gets harder to maintain and frequently produces verbose files. Tradition projects or environments that need strong external configuration are the ideal candidates for XML configuration.

Configuration of Annotation

By directly integrating data into the source code, annotation configuration streamlines the procedure. Spring can automatically identify and link beans at runtime with annotations like @Component, @Autowired, and @Qualifier. XML is kept as little as possible, frequently to enable component scanning. Compared to big XML files, this method speeds up development and reduces error rates. However, if modifications must be performed without modifying the source code, the configuration may become less flexible because it is included within the code.

Java-Based Configuration

A more newer option is Java-Based Configuration, in which beans and their dependencies are expressed as standard Java classes that have @Configuration and @Bean annotations. By utilizing Java's type safety and tooling support, this method increases configuration power and simplifies rewriting. It offers a streamlined and centralized method of managing beans, doing away with the necessity for XML altogether. This method, which combines readability and flexibility, is currently the standard in Spring Boot and modern Spring applications.

Comparation

The objective of defining and maintaining beans in the Spring IoC container is accomplished by all three methods. Java-based configuration is the preferable option nowadays because it combines readability, type safety, and contemporary techniques; annotations provide short but blend configuration with code; and XML stresses separation but is verbose. Essentially, Spring has improved developer productivity and maintainability at every stage, moving from XML to annotations and then to Java configuration.