

**实验报告（工科）**

|  |  |
| --- | --- |
| **学 院：** |  |
| **课程名称：** |  |
| **专业班级：** |  |
| **学 号：** |  |
| **姓 名：** |  |
| **指导老师：** |  |
| **职 称：** |  |

年 月 日

教务处印制

实验项目列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 实验项目名称 | 成绩 | 指导老师 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

广州理工学院实验报告

系： 计算机科学与工程 专业： 计算机科学与技术 年级： 22级

姓名： 邓然仪 学号： 202204114301028 组 实验时间： 2025.3.7

指导教师签字： 成绩：

|  |
| --- |
| 实验项目名称：  **实验一 Spring Boot 项目构建以及单元测试与热部署应用**   * **实验目的和要求：**   按照教材 1.2 搭建一个 Spring Boot 项目，结合 1.4 小节的内容和案例，完成 Spring Boot 入门  程序的实现，并完成 Spring Boot 项目的单元测试和热部署配置，最后对把该项目打包成 jar 包并测  试运行得到结果。 |
| * 主要仪器设备：   （1）、计算机（要求：CPU 500MHz以上，内存256M以上，磁盘空间2G以上。  （2）、Java 开发和运行环境。 |
| * 实验内容和原理：  1. Spring Boot 的单元测试和热部署。   单元测试  Spring Boot 通过 spring-boot-starter-test 起步依赖，集成了常用的测试框架和工具，Spring Test 等，方便开发者编写和执行测试。  测试环境配置：使用 @SpringBootTest 注解可以加载完整的应用程序上下文，进行集成测试。  模拟对象：通过 @MockBean 和 @SpyBean 注解，可以在测试中创建模拟对象，隔离被测组件的外部依赖。  热部署  Spring Boot 提供了 spring-boot-devtools 模块，支持应用的热部署，提高开发效率。  类加载机制：spring-boot-devtools 通过自定义类加载器，监控 classpath 下的资源变化。当检测到资源发生变化时，触发应用上下文的重启。  在项目运行期间，修改代码并保存后，devtools 会自动重新加载修改后的类，而无需手动重启整个应用。  LiveReload 集成：devtools 集成了 LiveReload 功能，当检测到静态资源（如 HTML、CSS、JS 等）变化时，自动刷新浏览器，方便前端开发调试。   1. Spring Boot 的自动化配置原理。   Spring Boot 的自动化配置使开发者无需手动配置大量的 XML 或 Java 配置，应用即可根据依赖和环境自动配置所需的 Bean 和功能。  @EnableAutoConfiguration 注解：该注解告诉 Spring Boot 根据类路径中的依赖和自定义的配置，自动配置 Spring 应用程序。  spring.factories 文件：在 META-INF/spring.factories 文件中，列出了需要自动配置的类。Spring Boot 在启动时会读取该文件，根据条件加载相应的配置类。  条件注解：自动配置类通常使用条件注解（如 @ConditionalOnClass、@ConditionalOnMissingBean 等）来判断某些配置是否生效，从而确保在特定条件下才进行自动配置。 |
| * 操作方法和实验步骤（绘图）： |
| 实验数据记录、处理和分析： |
| 质疑、建议： |