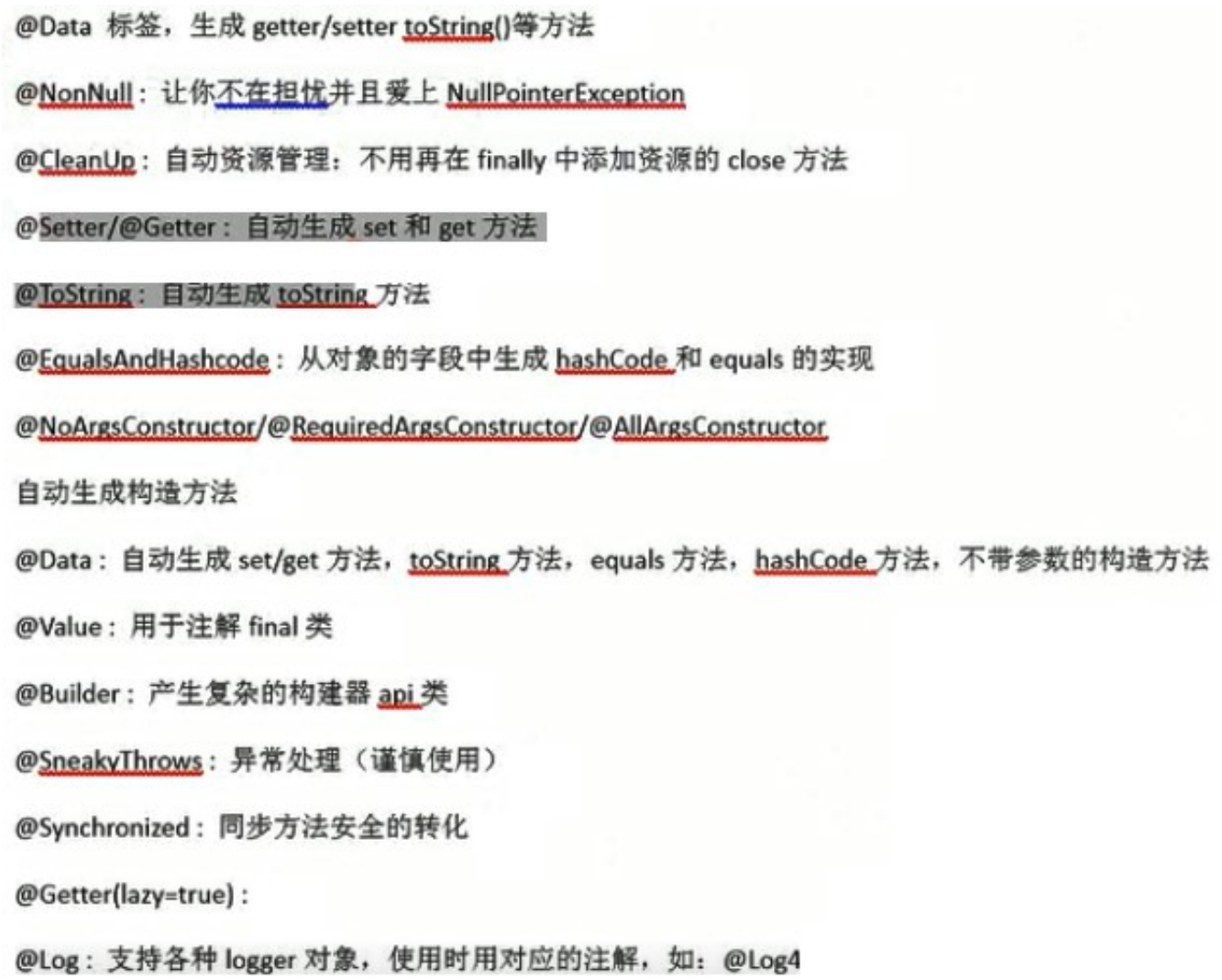
一、集成lombok让代码更简洁



lombok　底层使用字节码技术ASM修改字节码文件，生成get和set方法

1、引入依赖

```
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
```

2、@Getter @Setter　这两注解用于自动生成get/set方法

　@Slf4j　等同于日志　　private static final Logger logger = LoggerFactory.getLogger(当前Controller类.class);

　　使用：log.info(.....);

3、有可能还要安装lombok插件

二、@Async异步执行方法

启动类加上**@EnableAsync**,在Service服务类要执行的异步方法上加上**@Async**注解

那么加了@Async注解的类就是一个异步调用的方法，底层使用多线程技术。

@Async　相当于这个方法重新开辟了单独线程进行执行，原理是：AOP@Async技术在运行时创建一个单独线程进行执行。

```
@SpringBootApplication
@EnableAsync // 开启异步调用
public class JspApp {

    public static void main(String[] args) {
        SpringApplication.run(JspApp.class
    }

}
```

```
public class MemberService {

    // 添加用户的时候，会去发送邮件
    @Async // 相当于这个方法重新开辟了单独线程进行执行
    public String addMemberAndEmail() {
        log.info("2");
        try {
            Thread.sleep(5000);
        } catch (Exception e) {
            // TODO: handle exception
```