----------コードカバレッジツール？？

「STAR」、4つの英単語「Situation（状況）」「Task（課題）」「Action（行動）」「Result（結果）」

为了整合 xxx 业务(S)，我承担 xxx 角色，具体负责 xxx (T)。做了 xxx 事情(A)，最后产生了 xxx 结果

项目名称，背景，给哪个客户做，完成了基本的事情，做了多久，项目规模多大，用到哪些技术，数据库用什么，然后酌情简单说一下模块。重点突出背景，技术，数据库和其他和技术有关的信息。

我在XX公司做了XX外汇保证金交易平台，客户是XX银行，主要完成了挂盘，实盘成交，保证金杠杆成交等功能，数据库是Oracle，前台用到JS等技术，后台用到Java的SSH，几个人做了X个月。

    (介绍项目背景) （介绍项目技术，JD上列的尽量说全）（介绍项目管理工具）（最后抛出亮点）

我最近的项目是在xx公司做的线上商城项目，客户方是xx公司，项目组里是12个人，开发了八个月，最近还在开发，在其中我做了删除和更新订单模块的业务。

**三**

最新のプロジェクトは、xx 社のオンラインショッピングモールのプロジェクトです。チームのメンバーは 12 人です。開発期間は 8 か月で、今も開発中です。その中で、注文の削除と更新モジュールとを担当しました。

このプロジェクトは Spring Boot フレームワークを使用し、データベースはオラクルで、データベースへの接続に JPA が使用され、業務処理メソッドを記述するために制御層で @RequestMapping アノテーションが使用されます。ビジネスコードでは、@Autowired アノテーションを使用して IOC 形式でクラスを導入し、@Service アノテーションと @Component アノテーションも使用します。データベースからデータを取得するときは、@Entity アノテーションと @Table アノテーションを含むモデル クラスを使用してデータをマップします。このプロジェクトでは、トランザクションを定義するために @Transactional アノテーションも使用します。このアノテーションはサービス層で機能します。このプロジェクトでは、複数の Java コレクション クラスと例外処理クラスも使用します。

    这个项目用到了Spring Boot框架，数据库是Oracle，用JPA连接Oracle数据库，在控制层里用到了@RequestMapping注解来编写业务处理方法。在业务代码里，是用@Autowired注解来以IOC的方式引入类，还用到了@Service和@Component注解。当我们从数据库里拿到数据后，会用包含@Entity和@Table注解的模型类来映射数据。在这个项目里，我们还用到了@Transactional注解来定义事务，这个注解是作用在Service层上的，在这个项目里，我们还用到多个Java集合类和异常处理类。

  这个项目用到了敏捷开发模式，用Maven框架，用sonar管理质量，用git管理项目版本，用jenkins发布，用logback打日志， 我在这个项目里，除了写代码外，还有用Junit进行单元测试的经验。统计代码的测试覆盖度。

    在这个项目里，我用到了单例和工厂设计模式，在这个项目里，我做过JVM调优和数据库调优的的事情，在做项目时为了加深理解，我还看了ArrayList快速失效的底层代码，也看过HashMap关于HashCode以及put和get部分的底层源码。这个项目后期进度比较紧，我也能通过加班等手段保证项目能正常上线，而且一旦遇到线上问题，我也会尽量参与并解决。

**二**

我参与了将一家企业既存的系统从本地服务器迁移到AWS云上的项目。在这个项目中，我负责帮助设计和实施迁移计划，并对系统进行必要的修改和调整以适应AWS的技术栈。这个系统主要提供了一些基本的在线服务，包括用户登录、数据存储和处理、以及与第三方服务的接口。

我们使用了AWS的Elastic Beanstalk作为应用程序的主要托管服务，该服务可帮助我们轻松管理应用程序的部署和扩展。我们使用了DynamoDB作为我们的主要数据库，并使用Lambda函数作为应用程序的基本逻辑单元。Lambda函数通过使用API Gateway提供服务接口，并将数据存储到S3存储桶中。

为了更好地利用AWS的各种功能，我们采用了多种技术栈，包括Spring Boot框架、使用JPA连接数据库、使用Maven进行项目管理、使用Jenkins进行持续集成和部署、使用CloudFormation进行基础架构部署和管理、使用Elasticsearch进行数据搜索和分析等。

在这个项目中，我还负责了系统性能和安全的优化。我使用CloudWatch和AWS X-Ray对系统进行监控和调试，并使用AWS Identity and Access Management（IAM）来控制系统访问权限，确保安全性。

在迁移过程中，我们还面临了许多技术挑战，包括如何处理大量数据的迁移和如何兼容既存的代码库。我们使用AWS的Snowball服务来快速而可靠地迁移大量数据，使用CodeDeploy服务实现无缝的代码部署和更新，使用Lambda@Edge和CloudFront进行内容分发和加速。

最后，我们成功地将这个系统迁移到了AWS云上，取得了显著的性能和可靠性提升。我也从中获得了丰富的AWS使用经验和技能，包括使用AWS CLI和SDK进行自动化部署和管理，以及使用各种AWS服务进行系统优化和调试。