酒店管理系统

HMS (Hotel Management System)

单元测试与集成测试回顾

V1.0

钛4+小组 (Titanium Quartet)

蒋文荟 李佩瑶 李珍鸿 李一然

2016/12/18

目录

[一、单元测试 3](#_Toc469868804)

[1.1 测试概述 3](#_Toc469868805)

[1.2 测试用例的有效性分析 3](#_Toc469868806)

[1.3 采取的弥补手段 4](#_Toc469868807)

[1.4 度量数据 4](#_Toc469868808)

[二、集成测试 4](#_Toc469868809)

# 一、单元测试

## 1.1 测试概述

首先建立了数据层、逻辑层的Stub及Driver类，形成初步的体系结构框架，编写了简单的Junit测试用例。之后进行详细设计后，用Mock Object取代了逻辑层的Stub，以隔离外界的影响并体现具体的实现方案，对测试用例亦进行了相应的修改。

在构造过程中，逻辑层测试用例的改进和具体化在第二周开始展开，暴露了第一周的构造过程中出现的一些疏漏。测试的方法以随机测试和边界值测试为主，以保证逻辑层各个模块的稳健性。

## 1.2 测试用例的有效性分析

构造阶段之前，许多测试用例并不涉及具体的数据，只能测试逻辑主干是否能够正常执行。构造阶段开始后，许多测试用例仍然没有得到及时的改进。在增加了随机测试的使用频率，并且补充了一些针对边界值的测试用例后才得以精确地反映出复杂逻辑编写中的一些问题。

另外许多方法的执行有着一定的时序，测试时必须有调用的先后顺序，导致了测试用例与被测试方法不能一一对应的情况，为此调整了一些单元测试的命名规则。

## 1.3 采取的弥补手段

增加随机测试的使用频率；补充边界值测试；对复杂逻辑的部分分支应用白盒测试；应对接口的变更要即时修改，不能拖延。

## 1.4 度量数据

见Excel表格“测试度量数据”以及相应的HTML文档。

测试覆盖度的度量使用了EclEmma的Eclipse插件。

# 二、集成测试

体系结构初步形成时建立了Stub及Driver类，详细设计完成之后开发了Mock Object以隔离测试中环境的影响。详细设计的过程中完成了Jenkins的配置，并对整个项目（包括ClientModule, CommonModule, ServerModule三个模块）进行了应用。配置中发起构建的是两小时一次的定时器以及每十分钟检查一次Github的新Commit。

构造过程中，Jenkins的构建结果对提交不完善代码、逻辑层与数据层接口未及时同步、插件配置异常等问题作了及时的反映。

由于开始的一段时间数据库仅部署在了一名组员的机器上，集成测试对逻辑层、数据层的整体行为不能进行很好的测试，只能检查逻辑层。之后随着数据层的完善，集成测试得到了完善。