|  |  |
| --- | --- |
| **学 号：** | 0122010870323 |

****

**软件工程实践（一）**

|  |  |
| --- | --- |
| **学 院** | **计算机与人工智能学院** |
| **专 业** | **软件工程** |
| **班 级** | **软件2002** |
| **姓 名** | **周盟** |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2023 | 年 | 01 | 月 | 01 | 日 |

**目录**

[1 任务概述 1](#_Toc91495513)

[2 任务分析 1](#_Toc91495514)

[3 开发计划 1](#_Toc91495515)

[4 软件配置计划 1](#_Toc91495516)

[5 测试计划 1](#_Toc91495517)

[6 实施情况 1](#_Toc91495518)

[7 实施过程问题记录与分析 2](#_Toc91495519)

[8 任务总结 2](#_Toc91495520)

[9 参考文献 2](#_Toc91495521)

# 1 任务概述

任务目的

* 理解软件代码规范的重要性
* 理解代码变化对软件质量带来的影响
* 掌握基于Git的个人代码版本维护方法
* 掌握MarkDown文件编写方法

任务内容

* 阅读、理解和标注样例代码
* 分析和学习代码质量特征、设计方法和编程风格
* 运用所学方法，对开源代码进行标注
* 对样例工程进行简单功能扩充和维护

# 2 任务分析

1. 阅读和描述样例工程
   * fork样例工程，并clone到本地仓库；
   * 在本地开发环境上运行样例工程，理解样例工程的代码逻辑；
   * 精读样例工程软件代码，描述代码结构及部件组成；
   * 以UML图描述样例工程的组成及结构图（类及类之间的关系）
     + 可结合markdown语法和mermaid插件绘制所需图形
2. 标注样例工程中的代码
   * 基于javadoc规范标注代码，对包、类、方法、代码片段、参数和语句等代码层次进行注释（可参考Game类的标注样例）；
   * 注释后的代码提交到本地代码库后，同步推送到远程代码仓库；
   * 可参考ESLint、github/super-linter等开发插件了解关于代码规范的相关知识；
3. 扩充和维护样例工程
   * 对样例代码中的功能设计进行分析，找出若干设计缺陷和改进点，并进行修正或扩充，并集成到工程代码中；
   * 可借助代码质量分析工具或代码规范检查工具对代码质量进行分析，发现潜在问题；

# 3 开发计划

项目分析示例代码阅读阶段主要使用plantUML进行画类图以及做分析，项目维护和开发阶段主要使用IntelliJ IDEA 开发工具进行开发，在整个过程中使用git做代码的版本控制，使用Maven做依赖管理。

# 4 软件配置计划

本项目有两个分支：master和dev，主要在dev分支上进行代码开发工作，dev分支的代码测试通过没问题以后会合并到master分支。项目代码托管在github平台，本地开发每个阶段完成后将代码push到github平台。

# 5 测试计划

本项目中的测试部分主要对command改进部分和Room类的扩充部分进行了相关的测试，主要使用Junit单元测试工具进行测试代码的编写。

# 6 实施情况

**代码维护：**

由于在Game类的processCommand()方法中，当用户输入的命令被辨认出来以后，有一系列的if语句用来分派程序到不同的地方去执行。从面向对象的设计原则来看，这种解决方案不太好，因为每当要加入一个新的命令时，就得在这一堆if语句中再加入一个if分支，最终会导致这个方法的代码膨胀得极其臃肿。

**我的解决方案是：**

定义一个Command接口，在该接口中定义一个processCommand方法，项目中所有的命令类都需要实现该接口，而且项目中所有的命令类的命名必须遵循一定的规范：命令的首字母大写后面跟Command关键字 eg：go命令的命令类的命名：GoCommand。为了方便和管理项目中的命令我定义了一个Cmd注解，项目中所有的命令类都必须加上@Cmd注解，并在注解中标明该命令的名称。我在CommandWords类中写了一个静态代码块，当项目启动以来以后，会扫描项目中所有加了@Cmd注解的类将@Cmd注解的value值作为key，将该类的全类名作为value值加入到一个HashMap中。改进Parser类中的getCommand方法，在该方法中根据用户输入的命令关键字，从HashMap中获取该命令对应的命令类的全类名，再利用Java中的反射技术，根据类名创造出该类的对象并返回。此时Game类中的processCommand只需要调用Parser的getCommand方法返回的对象的接口方法processCommand便可以实现相应的功能而不必再做复杂的if判断。

**项目扩充**：

1. 扩充了项目中的Room类：在项目中新建了一个Item类用于表示每个场景中的物品信息，向Room类中添加了一个ArrayList用于保存该场景中的物品信息，增加了一个addGoods方法用于向该场景添加物品。

2. 扩充了Room类：使一些房间具有特殊的功能，当玩家进入该房间后会被随机传送到下一个房间。

3. 向项目中添加了back命令：当玩家输入该命令的时候可以返回到上一个房间。

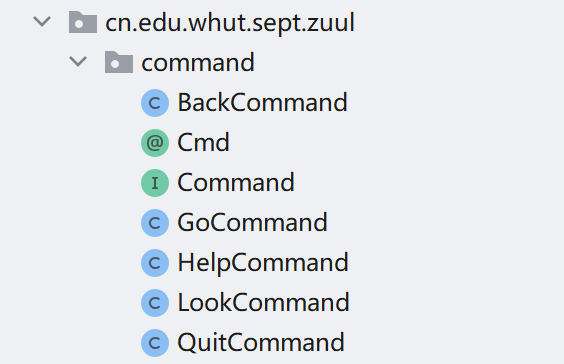
4. 向项目中添加了look命令：当玩家输入该命令后可以查看当前房间中的物品信息。

**标注**：

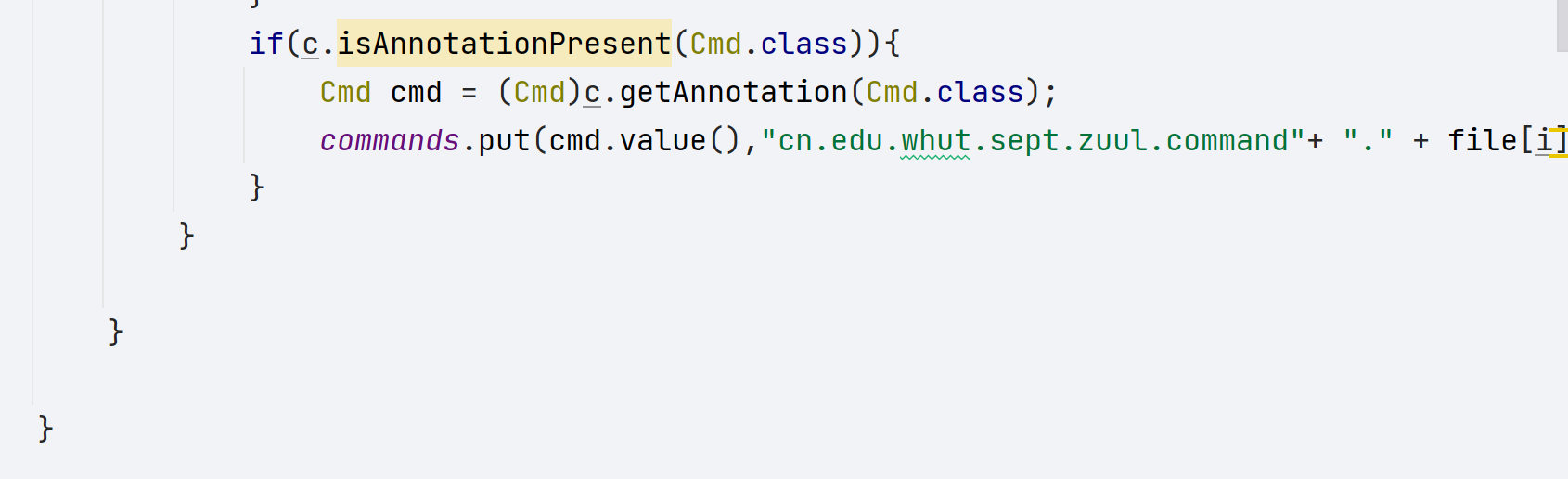
给项目中的所有代码添加了符合JavaDoc 规范的代码注释

# 7 实施过程问题记录与分析

Command模块的设计改进：

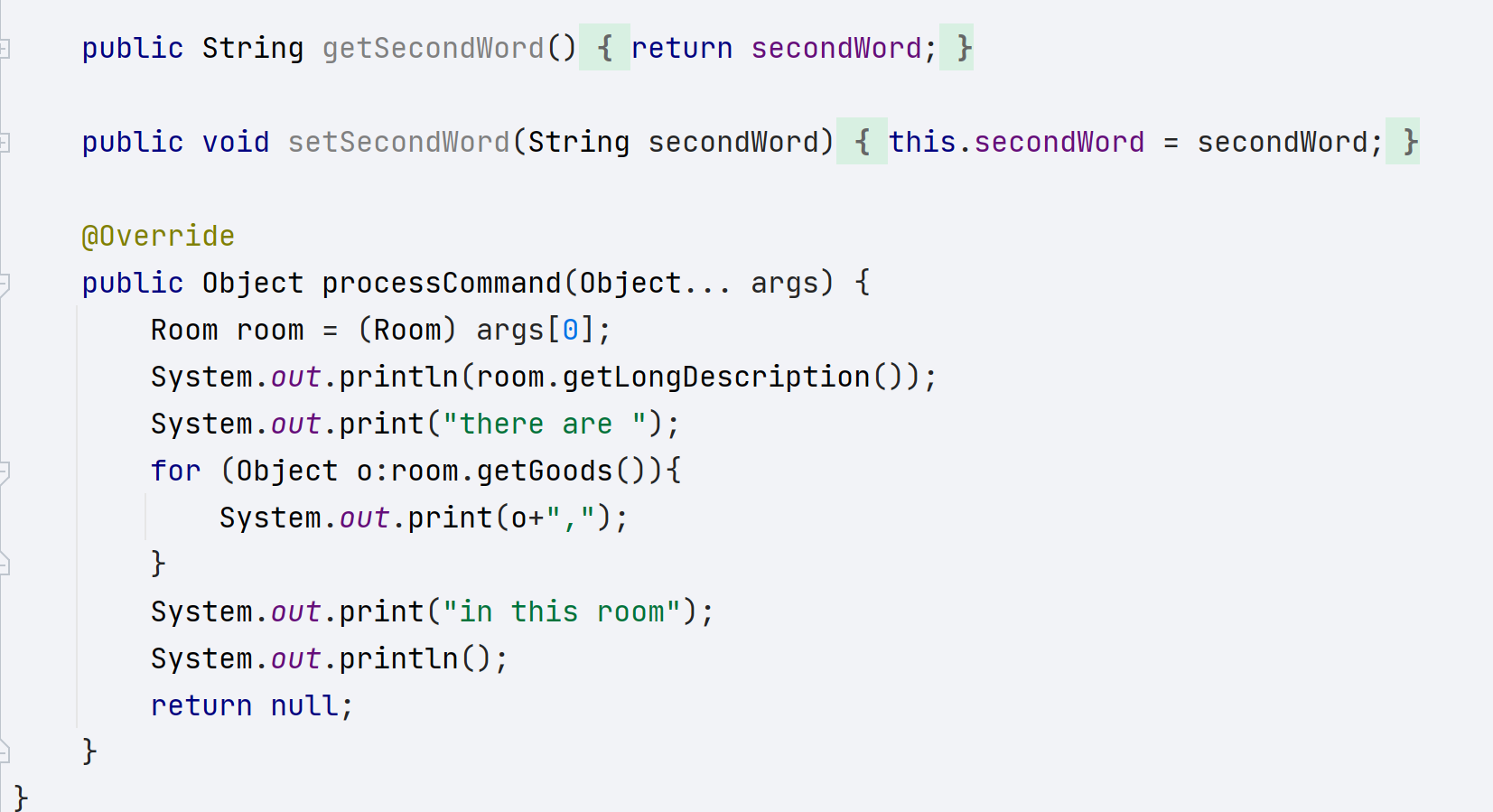






命令的扩充：

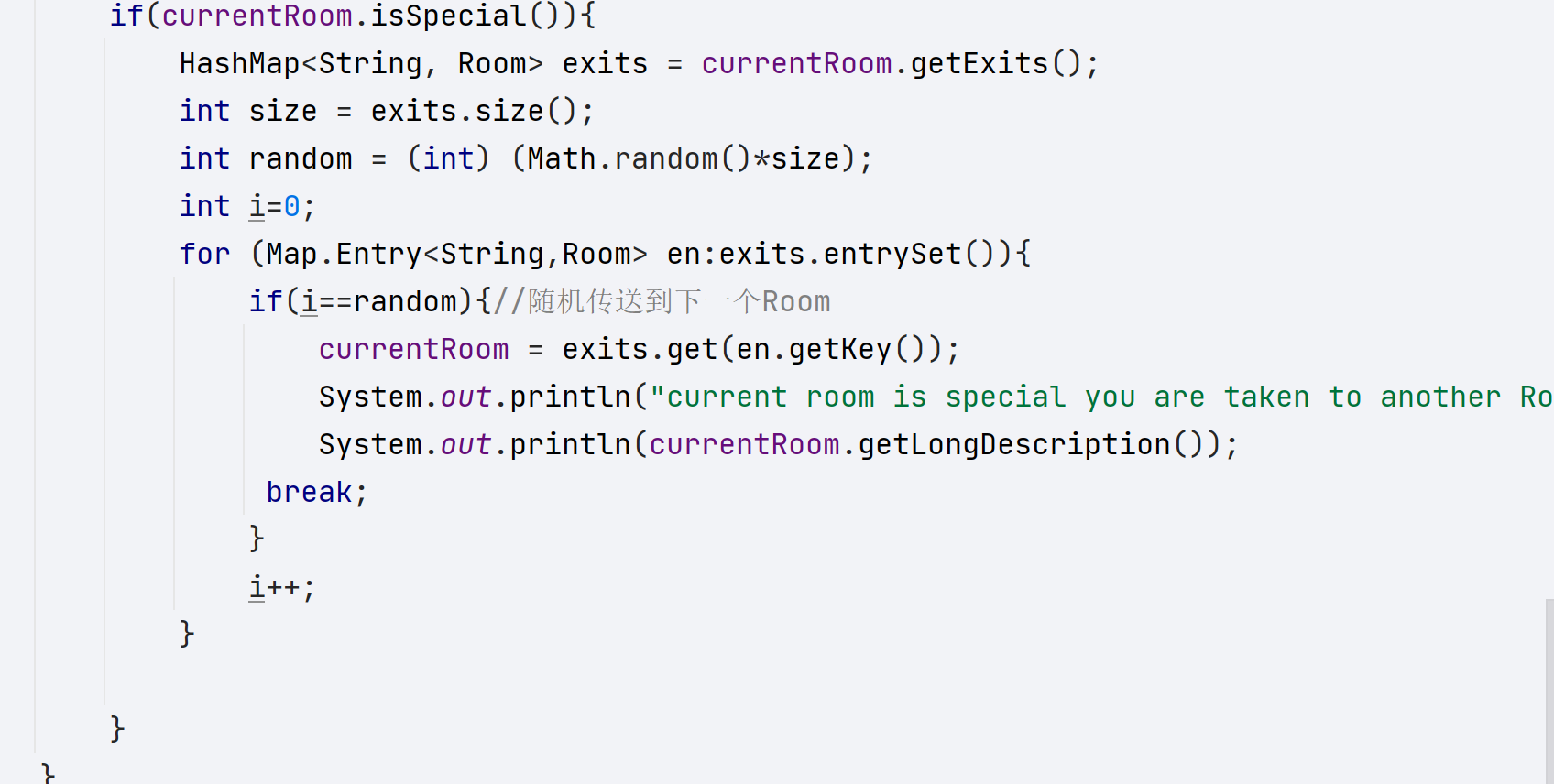




Room类的扩充：







# 8 任务总结

本次实验我看懂了样例代码，完成了对项目中的代码添加注释的工作，通过这次实验我明白了软件代码规范的重要性、好的架构设计对于代码维护和功能扩展的重要性 、掌握了基于Git的个人代码的版本维护方法、掌握了Markdown文件的编写方法。

**《软件工程实践（一）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | | **周盟** | **学 号** | **0122010870323** | |
| **专业、班级** | | **软件2002** | | | |
| **成绩评定：** | | | | | |
| 评价内容 | | | 满分 | 实得分 | |
| 得分 | 小计 |
| 实践任务  完成情况 | 软件项目设计、改进与扩充 | | 20 |  |  |
| 个人软件过程与项目管理 | | 15 |  |
| 代码版本管理 | | 25 |  |
| 代码注释与编码规范 | | 25 |  |
| 单元测试 | | 15 |  |
| 实践报告  总评情况 | 学习态度与考勤 | | 10 |  |  |
| 报告格式的规范性 | | 10 |  |
| 报告的逻辑结构与语言表达 | | 15 |  |
| 实践内容的正确性与合理性 | | 60 |  |
| 文献引用及标注 | | 5 |  |
| 总分 | | | 100 |  |  |
| **最终评定成绩（以优、良、中、及格、不及格评定）** | | | | | |

**指导教师签字：**

**年 月 日**