|  |  |
| --- | --- |
| **学 号：** |  |

****

**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** |  |
| **姓 名** |  |
| **编 号** |  |
| **指导教师** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 年 |  | 月 |  | 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

主要内容：

1、阅读、理解和标注样例代码

2、分析和学习代码质量特征、设计方法和编程风格

3、运用所学方法，对开源代码进行标注

4、对样例工程进行简单功能扩充和维护

任务目的：

1、理解软件代码规范的重要性

2、理解代码变化对软件质量带来的影响

3、掌握基于Git的个人代码版本维护方法

4、掌握MarkDown文件编写方法

# 2 任务分析

工作重点：

阅读、理解源代码，对源代码难点进行标注。寻找源码中代码质量不好的地方进行改进优化。完成对源代码中功能扩充部分的完成。

技术方案：

本实验主要运用Java面向对象的技术，技术难点不高，具有可行性。本实验扩充部分主要实现前三点，增加物品和back返回设计，增加物品可以通过新增物品类，在Room类中增加对物品类的使用即可，back返回通过利用栈保存房间信息，进行弹出即可进行切换当前房间。

工具使用方案：采用git bash进行git版本管理，采用IDEA进行java代码编写，采用Junit进行单元测试编写，采用Qaplug进行java代码规范审计，采用typora进行markdown编写， 采用mermaid和Visio进行UML绘制。

重难点分析：源码优化改进、新功能的添加。

# 3 开发计划

主要软件：

代码管理：git bash

代码编写、分析、测试：IDEA

Markdown：typora

UML：Visio

主要管理方法：指定项目完成时间表，按照时间表完成相应的工作。

# 4 软件配置计划

编码规范：文件内容编码采用GBK编码。

命名规范：java文件命名以主类名，开头字母为大写。

分支管理规范：代码保存在master分支上。

提交规范：每次提交通过comment添加备注，上传到master分支上。

# 5 测试计划

本次开发主要新增了Goods类，主要对Goods类进行单元测试。使用IDEA的插件JUnit进行单元测试的编写。针对Goods类中的函数编写特定的单元测试案例进行测试。

JUnit规则：

1、测试方法必须使用 @Test 修饰

2、测试方法必须使用 public void 进行修饰，不能带参数

3、一般使用单元测试会新建一个 test 目录存放测试代码，在生产部署的时候只需要将 test 目录下代码删除即可

4、测试代码的包应该和被测试代码包结构保持一致

5、测试单元中的每个方法必须可以独立测试，方法间不能有任何依赖

6、测试类一般使用 Test 作为类名的后缀

7、测试方法使一般用 test 作为方法名的前缀

在对源代码分析的情况下，发现以下一些问题并给出解决方案。

1、 避免使用像“HashMap”这样的实现类型；而应使用接口。在Room类中不应该使用private HashMap<String, Room> exits;，应该用private Map<String, Room> exits;代替。在Room类的Room(String description)中使用exits = new HashMap<String, Room>()进行初始化。

2、 在Parser类中的getCommand()方法中，定义了Scanner tokenizer读取键盘输入流的输入，但是在使用后没有对Scanner进行关闭，应该在后面加上tokenizer.close();对输入流进行关闭。

3、 CommandWord类中的常量名validCommands必须匹配表达式： '^[A-Z][A-Z0-9]\*(\_[A-Z0-9]+)\*$' ，更换成VALID\_COMMANDS。

4、 CommandWord类中的isCommand(String aString)中使用了没有花括号的if语句，加上花括号。

5、 在Room类中getExitString()方法在for循环中使用了+ 连接字符串，效率低下，缓存StringBuffer使用append方法连接。

# 6 实施情况

**完成情况：**

本实验的UML图如下：



本实验主要完成了前三项扩充任务，即:

1、扩展游戏，使得一个房间里可以存放任意数量的物件，每个物件可以有一个描述和一个重量值，玩家进入一个房间后，可以通过“look”命令查看当前房间的信息以及房间内的所有物品信息；

2、在游戏中实现一个“back”命令，玩家输入该命令后会把玩家带回上一个房间；

3、在游戏中实现一个更高级的“back”命令，重复使用它就可以逐层回退几个房间，直到把玩家带回到游戏的起点；

**运行结果：**

1、“look”命令可以查看房间信息和物品信息

**文本

描述已自动生成**

2、“back”命令把玩家带回上一个房间

**文本

描述已自动生成**

3、多次“back”命令将玩家带回起点

文本

描述已自动生成

**测试情况：**

**文本

描述已自动生成**

对Goods类的测试符合预期。

**软件代码提交情况：**

文本, 聊天或短信

描述已自动生成

首先git add选择需要提交的文件，然后git commit -m "提交信息" ，将文件提交到本地仓库，提交信息必须填写，最后git push -u origin master，将代码提交到github。

# 7 实施过程问题记录与分析

当实现Goods类的时候，构造函数的时候，参数名称与成员变量相同，之间赋值导致出现报错，要通过self调用成员变量进行赋值，这样子才可以对成员变量实现赋值。

当实现back功能的时候，多次使用back的时候会出现报错，发现在栈弹出的时候没有做是否栈空判断，当栈为空的时候还进行弹出导致错误，加上判断后报错解决。

# 8 任务总结

本次项目大体上算完成了，经过这次项目，我又学到了很多的知识，尤其是运用软件工程知识去解决实际问题的能力又得到了很大的提高。尽管所设计的程序比较简单，但是程序中的每个功能的实现都经过了自己的一定认真的思考。由于把所学的知识做到很灵活的运用在实际中，对于现在的我还有一定的难度，故而，在设计的每个的环节中，都会多多少少的遇到一些问题，这就促使我去查阅各种资料，或者请教老师和同学，极大的提高我的专业素养，同时对在有些书上的理论知识在实际的运用，以及应该注意的问题有了比较深刻的认识。当然，仍然有些理论问题不是很好的去运用，或者是了解的不深刻。这就使的在实践中的所设计的程序有些功能不是自己所想的那样的完美。不过，既然发现了问题的所在，我就不会放弃的，直至问题得到解决。这些也同时反映出了平时运用知识去实践这方面做的太少。太缺乏经验。对所设计中将所面临的问题认识不够。也使我充分的认识到在以后也应该怎样去实践，怎样去学习软件工程，以及如何去运用软件工程去解决实际问题。

# 9 参考文献

[1] git文档 (<https://git-scm.com/docs/gittutorial>)

[2] mermaid教程(<https://github.com/mermaid-js/mermaid> )

**《软件工程实践（一）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | |  | **学 号** |  | |
| **专业、班级** | |  | | | |
| **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**