|  |  |
| --- | --- |
| **学 号：** | 0122010870312 |

****

**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** | **李成吉** |
| **编 号** | **0122010870312** |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2022 | 年 | 12 | 月 | 30 | 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

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

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

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

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；

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；

提示：样例工程的代码结构存在一些可以改进的功能点，可参考下列说明进行改进：

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。如何改进程序中的这个设计，使得命令的处理更模块化，且新命令的加入能更轻松？请描述你的解决思路，并对你的解决方案进行实现和测试。

功能扩充点

样例工程“world-of-zuul”具备最基本的程序功能，该项目具有极大的扩展空间，各位同学可选择或自行设计系统结构优化或功能扩充需求，完成3项左右的功能扩充实现；

可供参考的结构优化或功能扩充项包括但不限于以下内容：

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；

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；

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；

在游戏中增加具有传输功能的房间，每当玩家进入这个房间，就会被随机地传输到另一个房间；

在游戏中新建一个独立的Player类用来表示玩家，并实现下列功能需求：

一个玩家对象应该保存玩家的姓名等基本信息，也应该保存玩家当前所在的房间；

玩家可以随身携带任意数量的物件，但随身物品的总重量不能操过某个上限值；

在游戏中增加两个新的命令“take”和“drop”，使得玩家可以拾取房间内的指定物品或丢弃身上携带的某件或全部物品，当拾取新的物件时超过了玩家可携带的重量上限，系统应给出提示；

在游戏中增加一个新的命令“items”, 可以打印出当前房间内所有的物件及总重量，以及玩家随身携带的所有物件及总重量；

在某个或某些房间中随机增加一个magic cookie（魔法饼干）物件，并增加一个“eat cookie”命令，如果玩家找到并吃掉魔法饼干，就可以增长玩家的负重能力；

扩充游戏基本架构，使其支持网络多人游戏模式，具备玩家登陆等功能；

为单机或网络版游戏增加图形化用户界面，用过可以通过图形化界面执行游戏功能；

可以为游戏增加数据库功能，用于保存游戏状态和用户设置；

......

编写测试用例

针对功能改进和扩充，在项目结构中编写单元测试用例，对代码执行单元测试；

以UML图表示的样例工程软件结构；

在所有源代码文件中完成源代码标注和扩充，并通过git提交到代码库；

# 2 任务分析

软件项目设计

主要内容：基于程序设计方法，结合样例或自选题目，完成开发案例的设计、功能点扩展和功能实现

实践重点：面向对象程序设计方法在实践工程项目中的应用

个人软件过程与项目管理

主要内容：基于软件工程开发流程，针对设计任务采用适当的个人软件过程与项目管理工具进行项目流程管理

实践重点：可利用Maven/Gradle等管理工具支持个人软件过程开发阶段的划分、软件项目依赖的管理

代码版本管理

主要内容：基于代码版本管理的概念与方法，对开发过程中代码仓库、工作空间、版本基线、代码提交与合并过程进行控制

实践重点：可利用Git工具支持个人代码版本管理

软件编码规范

主要内容：遵循开发语言编码规范，保持软件代码的可读性与可维护性

实践重点：可利用CodeStyle等代码规范检查工具进行规范性约束和检查

软件单元测试

主要内容：按照单元测试理论，在项目开发过程中通过测试用例实施单元测试

实践重点：可基于Junit框架实施单元测试和测试驱动开发

# 3 开发计划

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；

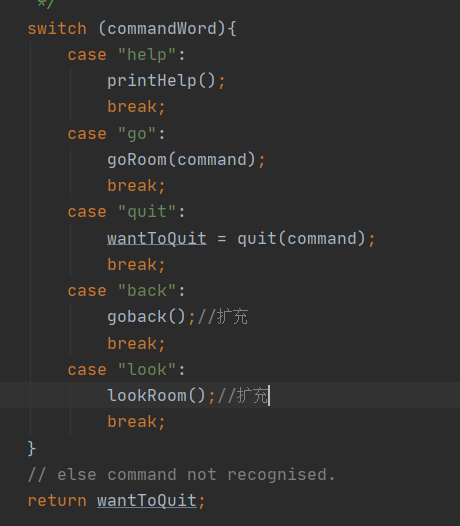
以UML图描述样例工程的组成及结构图（类及类之间的关系）

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；

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；

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；

对Game类中的processCommand方法，当用户输入的命令被识别出来之后，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。通过将if语句改为switch-case语句，可以防止当要新加一个命令时会新增一个if分支而使得代码臃肿。



样例工程“world-of-zuul”具备最基本的程序功能，该项目具有极大的扩展空间，各位同学可选择或自行设计系统结构优化或功能扩充需求，完成3项左右的功能扩充实现；

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；

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；

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；

在游戏中增加具有传输功能的房间，每当玩家进入这个房间，就会被随机地传输到另一个房间；

针对功能改进和扩充，在项目结构中编写单元测试用例，对代码执行单元测试。

通过idea使用junit对类中方法进行单元测试

对CommandWords类进行单元测试，测试方法：public void isCommand()

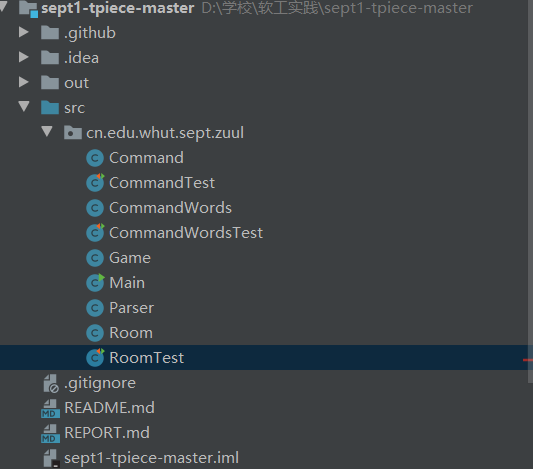
对Command类进行单元测试，测试方法：public void getCommandWord()，public void getSecondWord()，public void isUnknown()，public void hasSecondWord()

对Room类进行单元测试，测试方法：public void getShortDescription()，public void getLongDescription()，public void getExit()

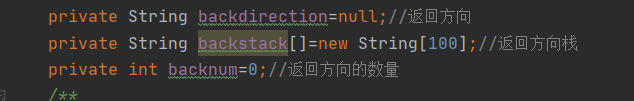
# 4 软件配置计划

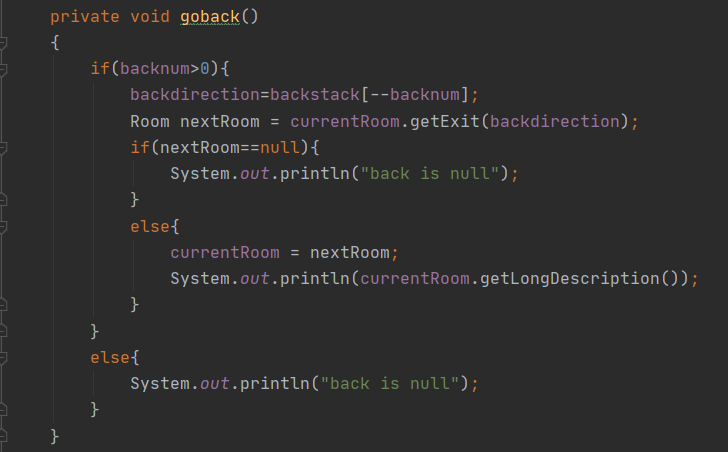
**对源代码文件及版本的编码规范、命名规范、分支管理规范、提交规范等内容进行约定说明**

代码目录：

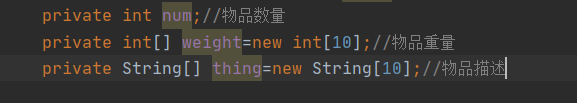


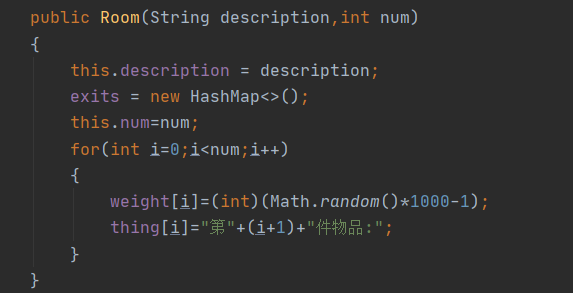
新增的back命令中的成员：

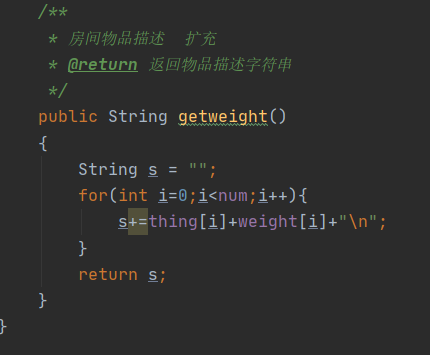




新增look命令中的成员：



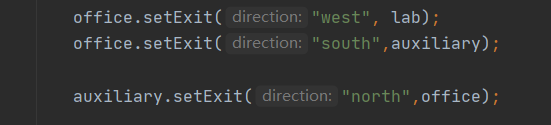






新增随机传送房间：auxiliary







# 5 测试计划

**对开发过程中涉及的单元测试及相关测试任务进行规划，对使用的测试工具、主要测试规范进行说明**

针对功能改进和扩充，在项目结构中编写单元测试用例，对代码执行单元测试。

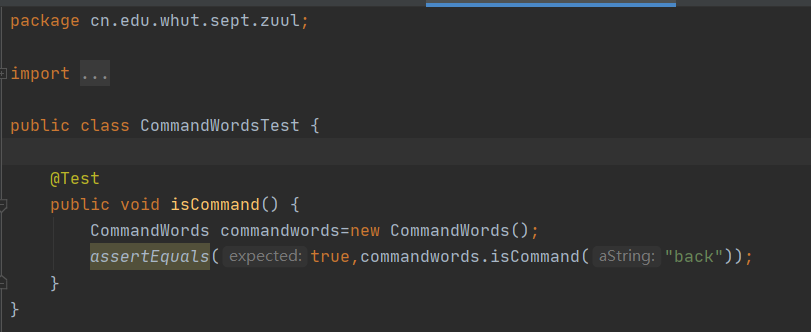
通过idea使用junit对类中方法进行单元测试

对CommandWords类进行单元测试，测试方法：public void isCommand()

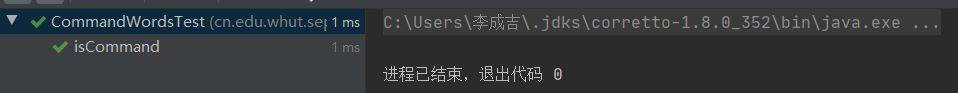
对Command类进行单元测试，测试方法：public void getCommandWord()，public void getSecondWord()，public void isUnknown()，public void hasSecondWord()

对Room类进行单元测试，测试方法：public void getShortDescription()，public void getLongDescription()，public void getExit()

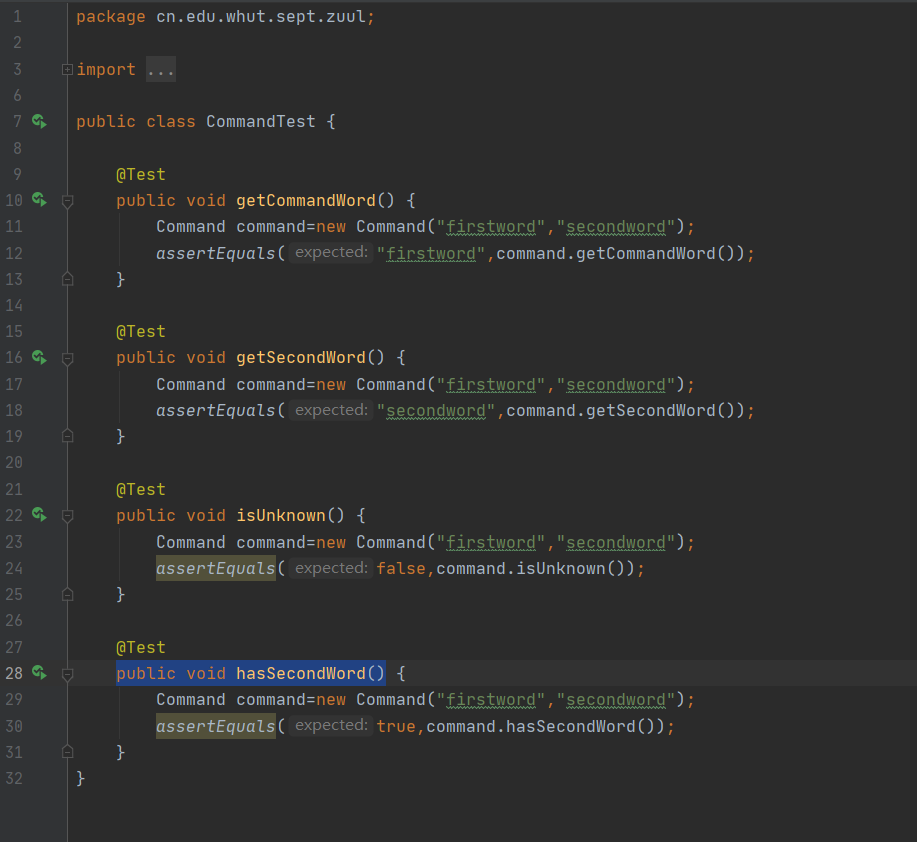
CommandWordsTest类：



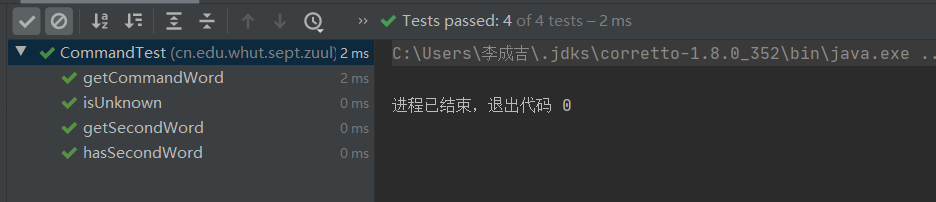
测试结果：



CommandTest类：



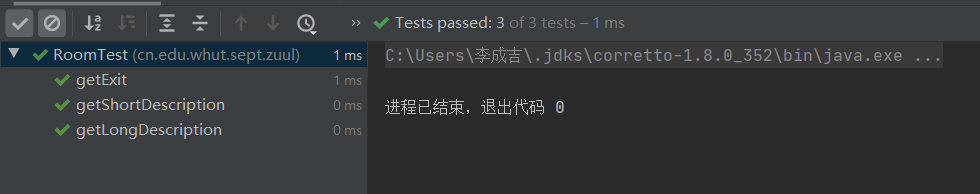
测试结果：



RoomTest类：



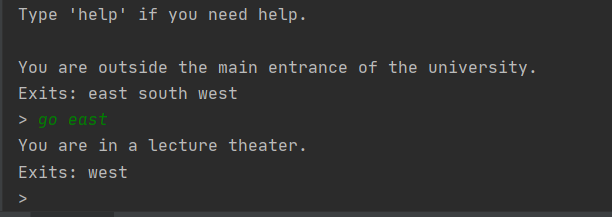
在对Room类中添加了房间中物品信息时，该类的构造函数也随之改变，同时对应的方法也相应改变



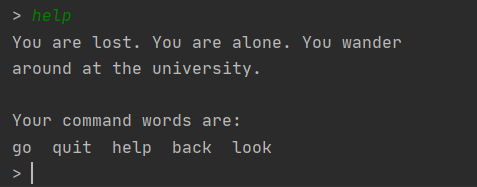
# 6 实施情况

**对项目开发的过程和结果进行说明，主要包括功能设计方案及完成情况、运行结果、测试情况与修正结果、软件代码提交情况进行说明**

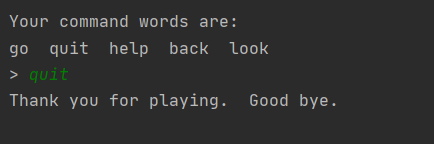
执行go命令：



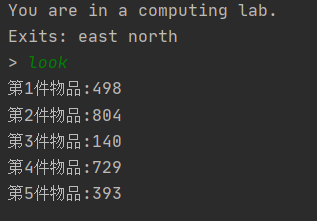
执行help命令：

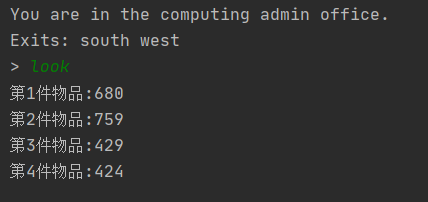


执行quit命令：



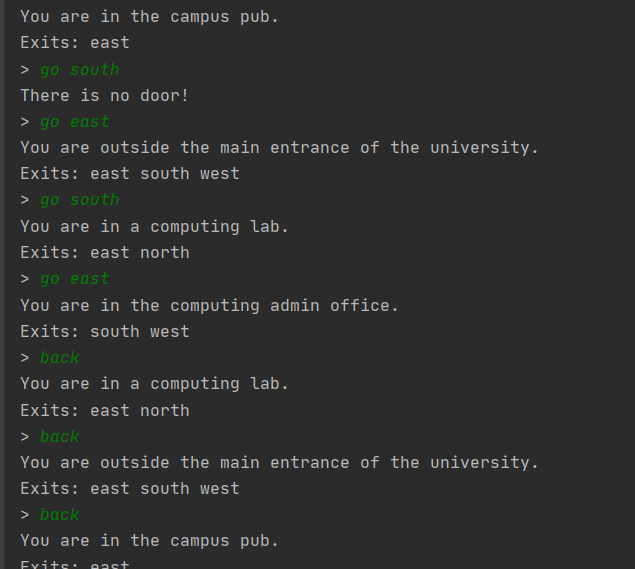
执行look命令：





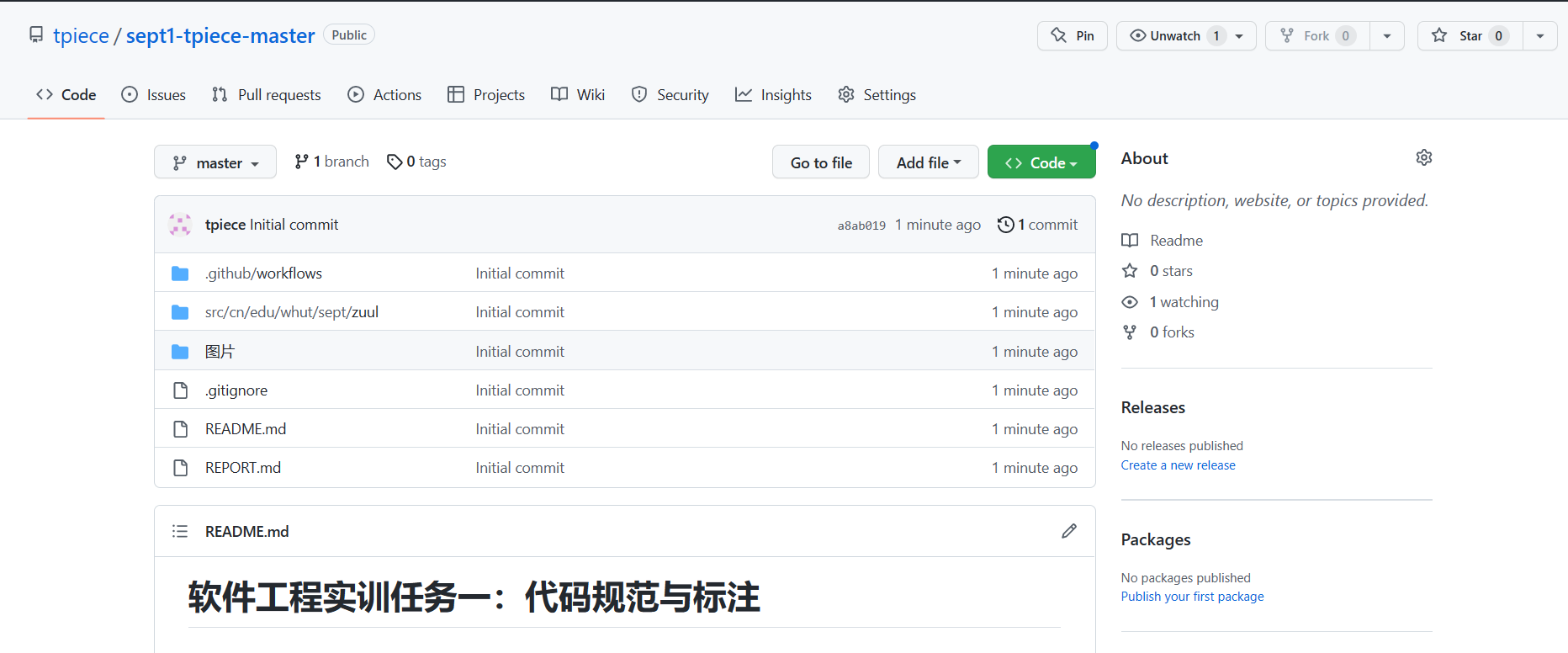
执行back命令：

实现高级的back命令可以逐层返回多个房间



项目提交到github

项目链接：[tpiece/sept1-tpiece-master (github.com)](https://github.com/tpiece/sept1-tpiece-master)



# 7 实施过程问题记录与分析

**对项目开发的过程中遇到的问题进行记录和分析**

在此次实验中，遇到了许多问题，比如在设置随机传送房间时传送到另一个房间后便无法使用back命令，便对随机传送进行修改，修改后的随机传送为从随机传送房间随机选择方向行走50次后到达某个房间。在以上方法更新后，又遇见了新的问题，通过随机传送后，可能再次进入随机传送房间，而再次进入后便不会进行传送，便对代码中的bug进行修改，进入随机传送房间后再传送不会进入该房间。实验中还有其他问题，比如对back操作进行修改时要运用到栈来存储返回的方向从而可以实现多级返回操作，直至返回起点。

# 8 任务总结

**对项目实践完成的总体情况，经验或者收获等进行总结**

通过此次实验，我对于代码规范和基于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，同时，此次实验中，我也学会了如何使用junit对代码进行单元测试。通过对于高级back命令的编写，也让我对于栈的知识有了一定的回顾。对于改进维护样例工程，，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（如SonarQube、ESLint等)对代码质量进行分析，发现潜在问题。在最后的代码提交过程中，我对于git的操作也更加深刻，也学会了如何将代码上传到github仓库中。我认为此次实验收获最大的是利用Maven/Gradle等管理工具支持个人软件过程开发阶段的划分、软件项目依赖的管理

# 9 参考文献

《Java设计模式》作者：耿祥义，张跃平 出版社：清华大学出版社 出版时间：2009年

《UML和模式应用(第三版)》作者：Larman 出版社：机械工业出版社 出版时间：2006年

《Essential Software Architecture》作者：Ian Gorton 出版社：Springer 出版时间：2006年

《企业应用架构模式》作者：（英）福勒，王怀民，周斌　译 出版社：机械工业出版社 出版时间：2010年

Metsker S J.Java设计模式.第二版.电子工业出版社.2012

**《软件工程实践（一）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | | **李成吉** | **学 号** | **0122010870312** | |
| **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**