|  |  |
| --- | --- |
| **学 号：** | **0122010870108** |

****

**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班** |
| **姓 名** | **王煜泽** |
| **编 号** |  |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2023 | 年 | 1 | 月 | 3 | 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

**任务要求：**

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；

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。

1. 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；
  + ......

1. 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；

# 2 任务分析

本次任务旨在以个人为开发单位，完成“world of zuul“游戏的开发。为了实现这一目标，需要阅读理解样例代码，对游戏程序的逻辑深入理解，才能基于样例代码实现

原有功能扩充和功能设计改进。

本次任务的难点在于需要掌握markdown语法基于mermaid插件绘制本次工程的UML图设计和绘制，使用github进行远程代码管理，使用Maven框架对软件项目进行管理，最后还需要使用junit和powermock等单元测试工具对项目功能进行单元测试。

# 3 开发计划

1、基于IntelliJ IDEA 2021.2.4（Ultimate Edition）工具进行本地工程开发。

2、基于typera工具使用markdown语法的mermaid语言绘制uml图

3、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。

4、使用github工具进行个人代码版本管理，对开发过程中代码仓库、版本基线、代码提交与合并过程进行控制，扩充和维护游戏工程。

5、使用junit和powermock对项目进行单元测试。

# 4 软件配置计划

本次项目共有四个分支和master主干。

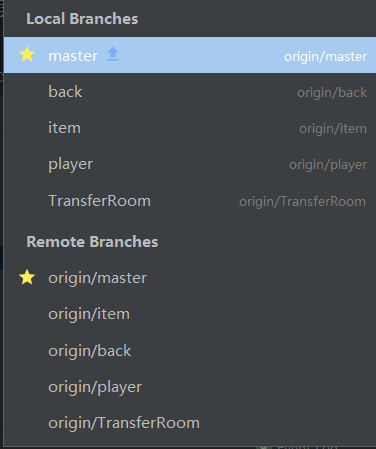
1、Item分支的任务是实现房间存放物件和物品展示的功能。实现指令look

2、back分支的任务是实现玩家指定回溯次数，带回到过去某个历史记录点。实现指令back

3、TransferRoom分支的任务是实现随机传送房间的功能。

4、Player分支的任务是实现玩家携带、拾取、丢弃和食用魔法饼干的功能。实现指令take，drop，items，eat cookie。

每个分支都会进行开发与测试，最后进行功能完善，过程中使用commit保存到本地仓库，最终将每个阶段结果push到github平台，确认无误后merge到master主干上。



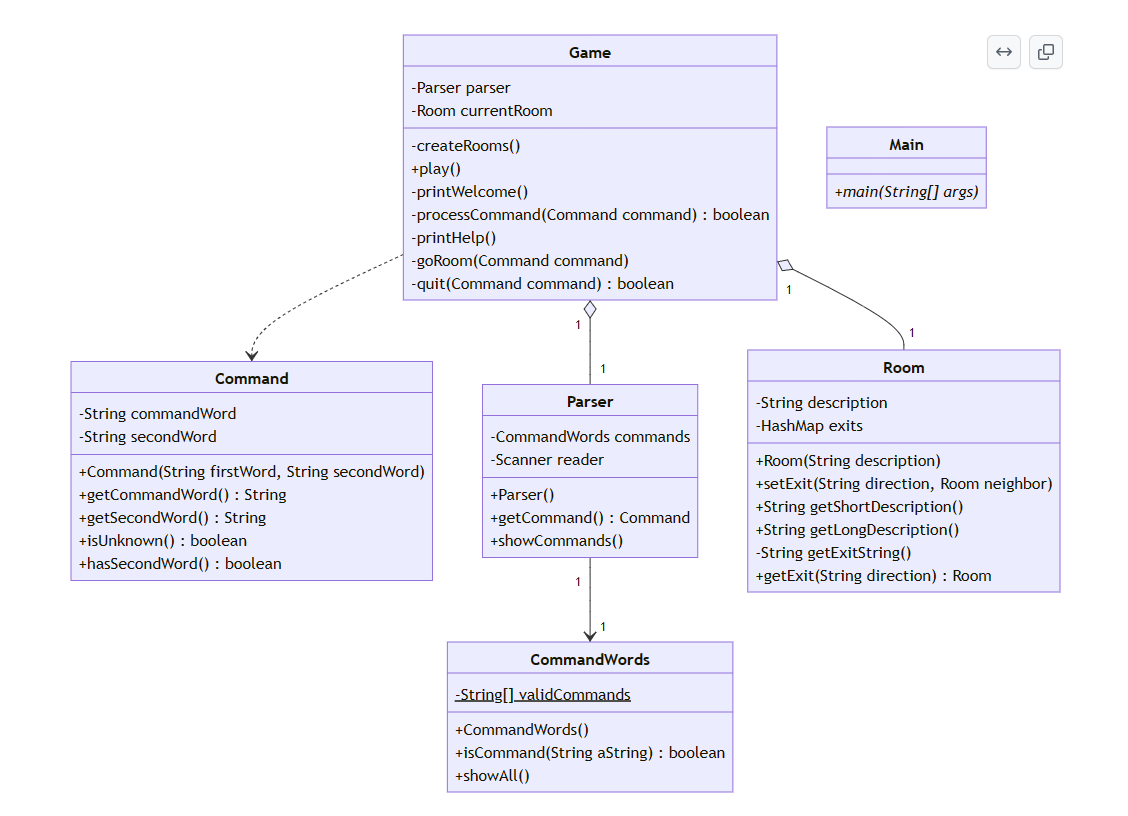
# 5 测试计划

本次实践的单元测试基于powermock和junit实现，主要针对Game类中执行back，take，drop，eat cookie进行测试，其他的指令主要是对（Player或Room）包含的item以及Room对象进行输出，只要规范好输出的内容和格式即可。

所以需要编写backTest，takeTest，dropTest，eat\_cookieTest四个测试方法。

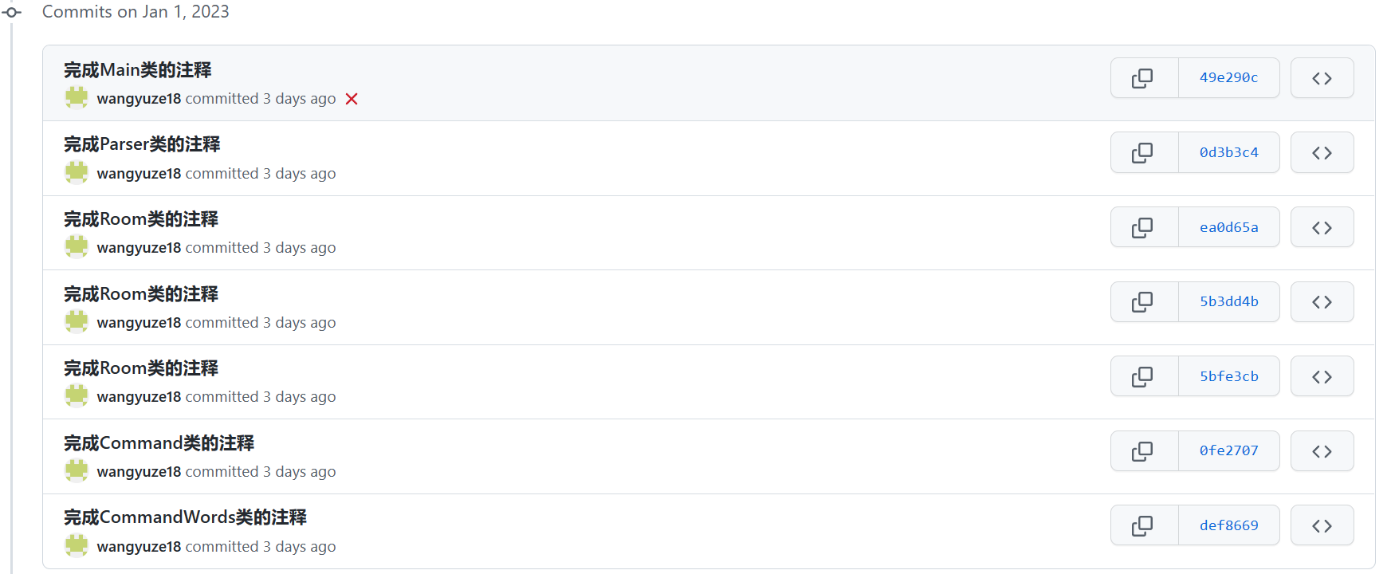
# 6 实施情况

**1、阅读和描述样例工程**



**2、标注样例工程代码**

具体内容可以在github上查看



**3、扩充和维护样例工程**

1、问题：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。

解决方案：使用switch语句来代替if else语句实现优化。

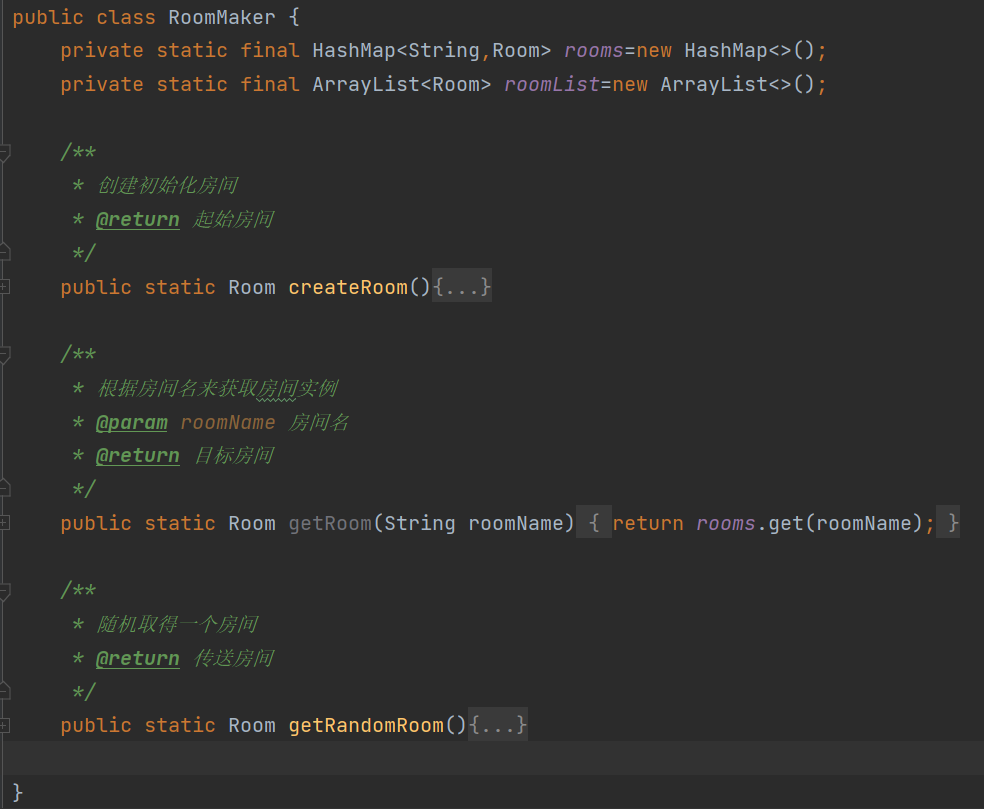




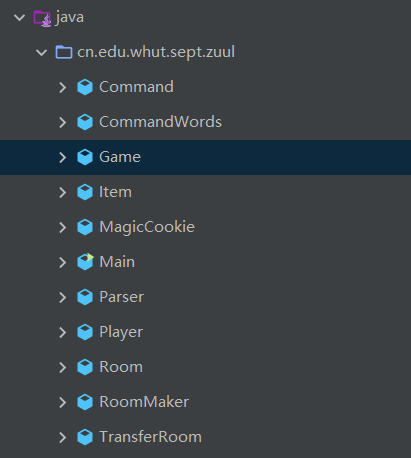


2、问题：在Game类的createRoom()方法中，当创建好游戏中房间后，不容易操作房间，且在后续地图复杂后或是随机生成功能实现后，Game类会变得十分臃肿

解决方案：创建RoomMaker工具类作为工厂，来创建生成Room并提供新的对房间操作方法。

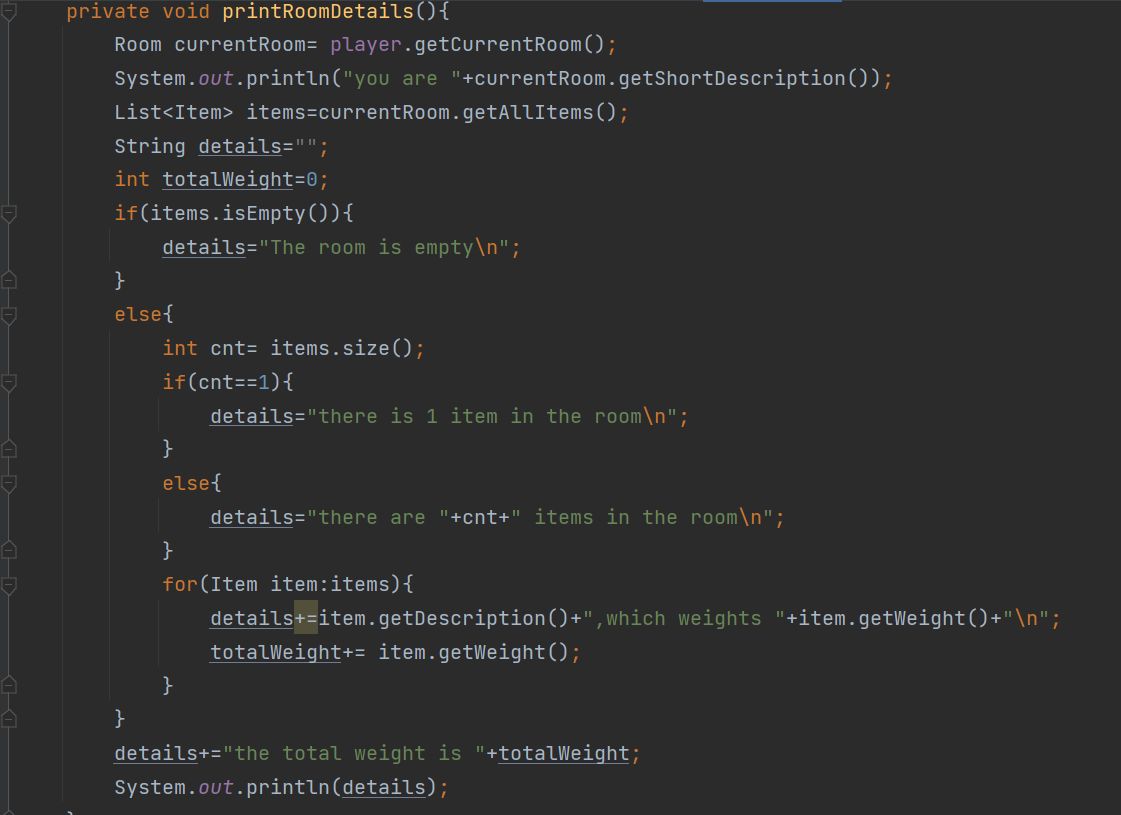


4、功能扩充点

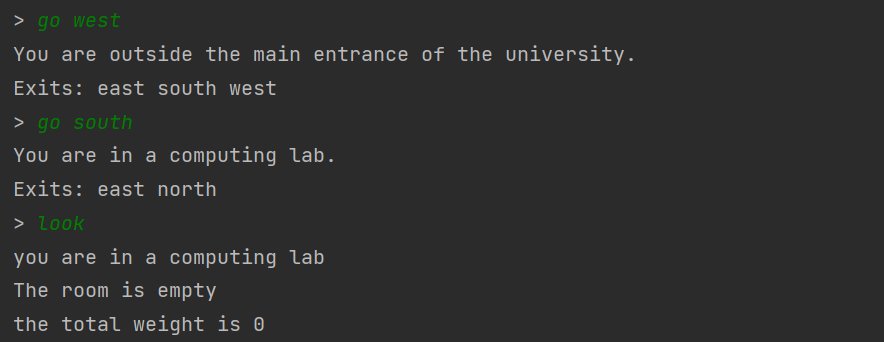
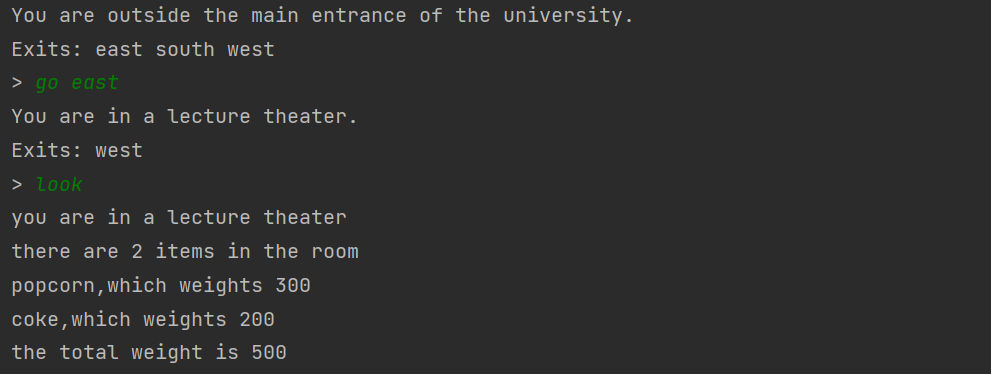


1、look命令的实现，可以查看当前房间内的物品

新增了item类来记录物品名和重量。Room类中新增了HashMap<String,Item>来存放物件。



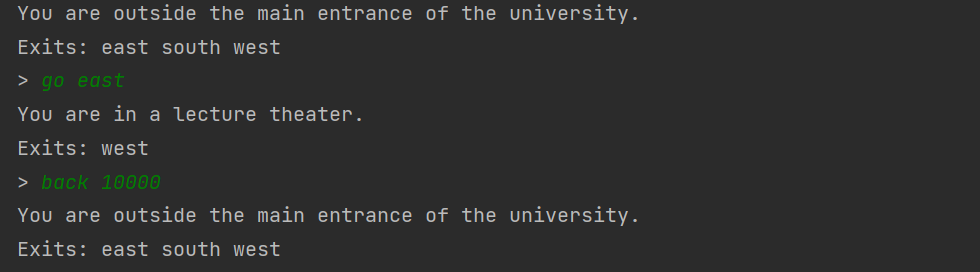
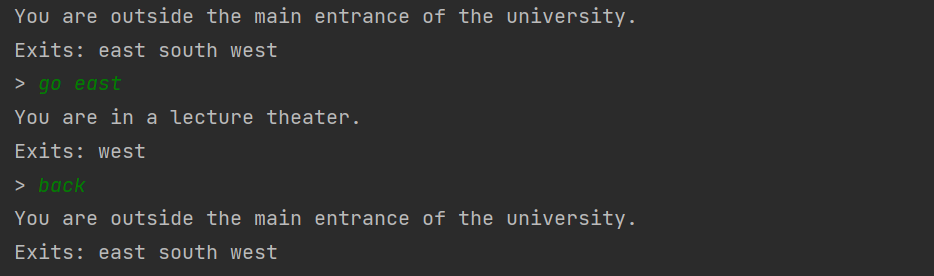
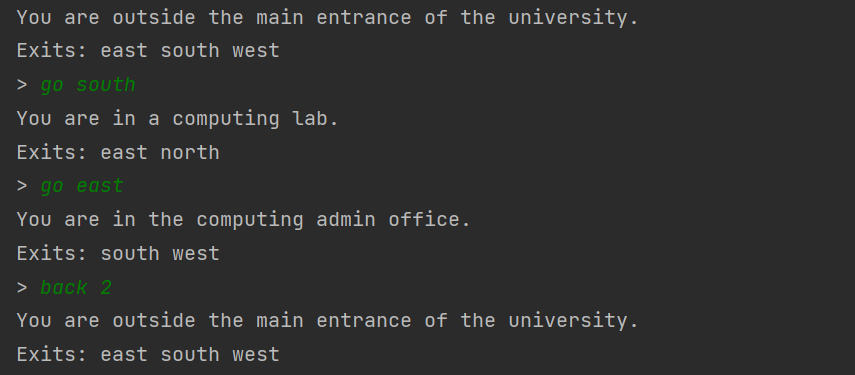
效果演示：



2、back指令的实现，可以回溯任意个房间直至到达起点



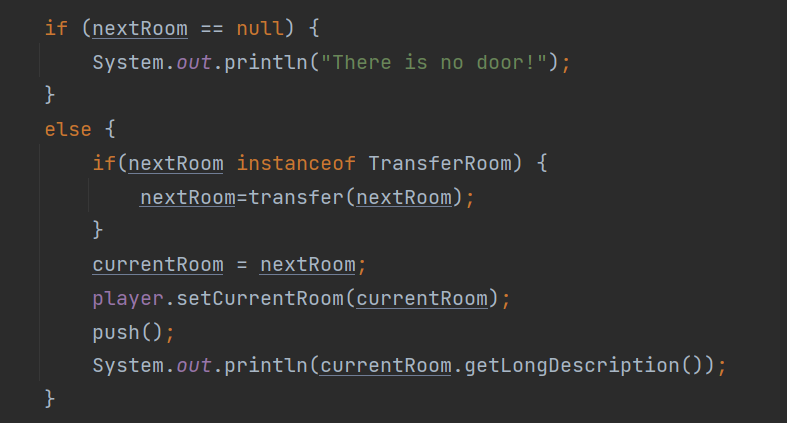
效果演示：

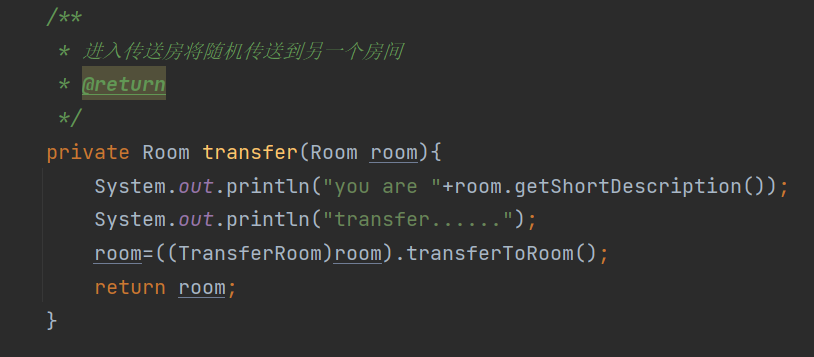


3、传输功能的房间实现，进入房间后会传送到任意其他房间

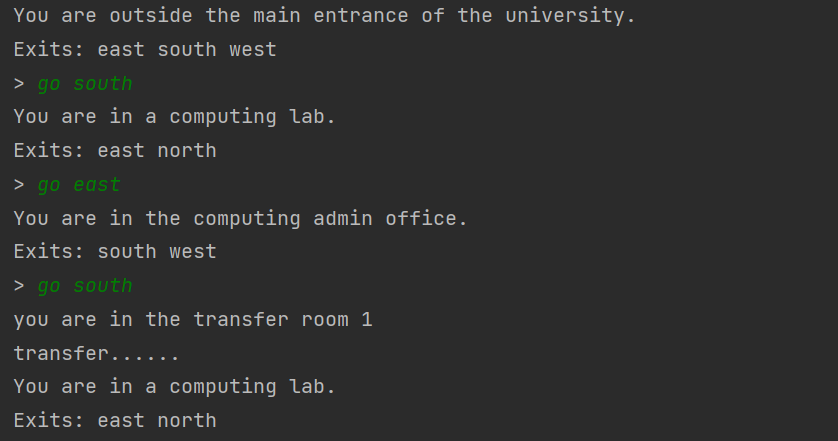
TranferRoom类继承自Room类，扩充了transferToRoom（）方法

goRoom方法：

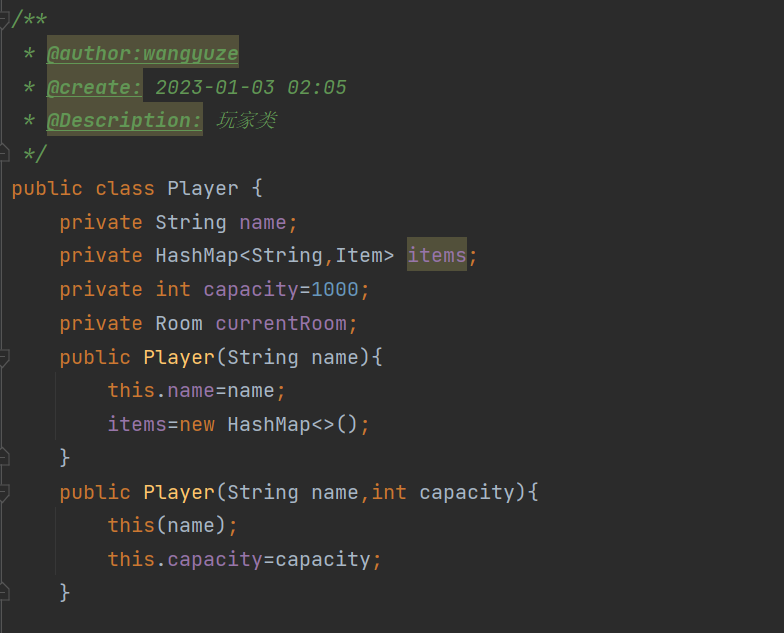




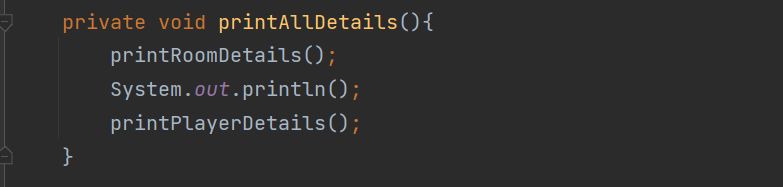
效果演示：

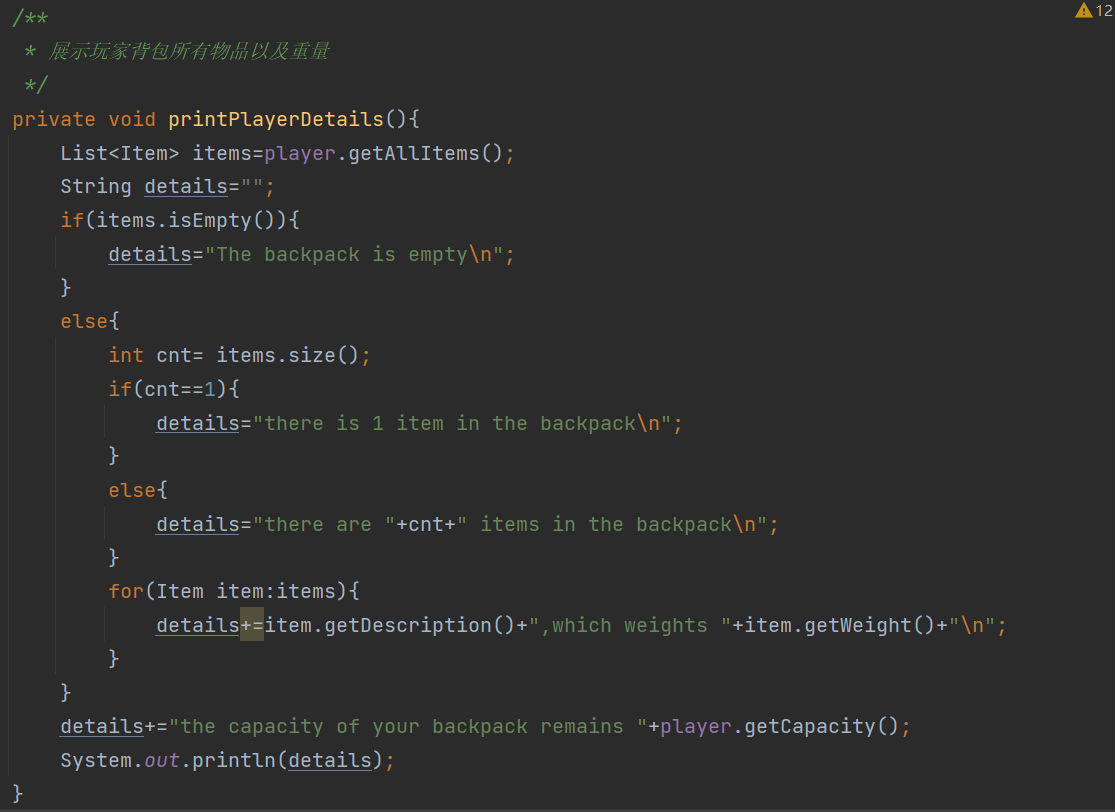


4、新增Player类

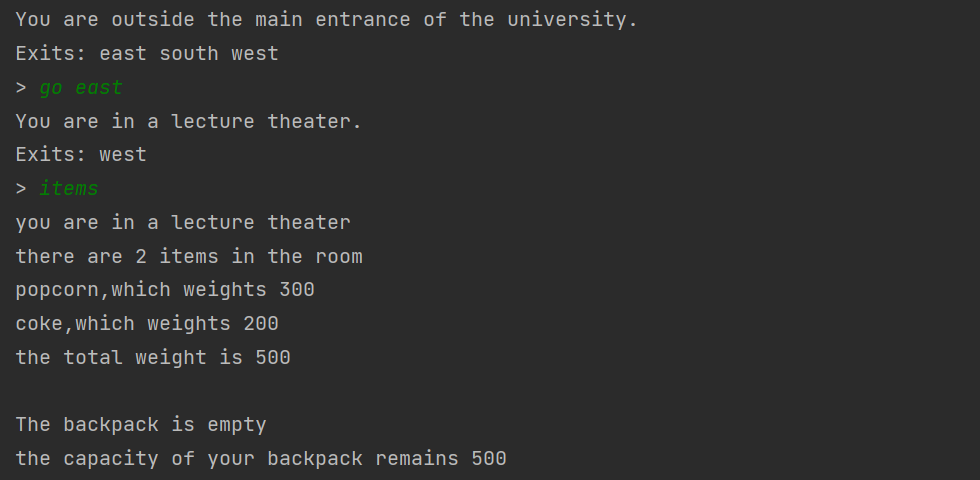


5、items指令的实现，查看当前房间内的物件和背包内的物件





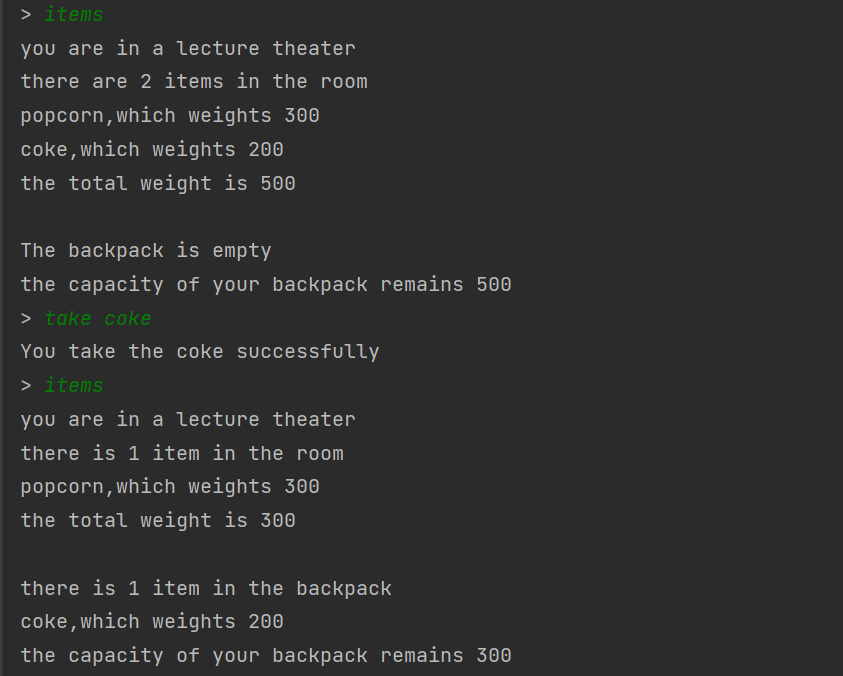
效果展示：



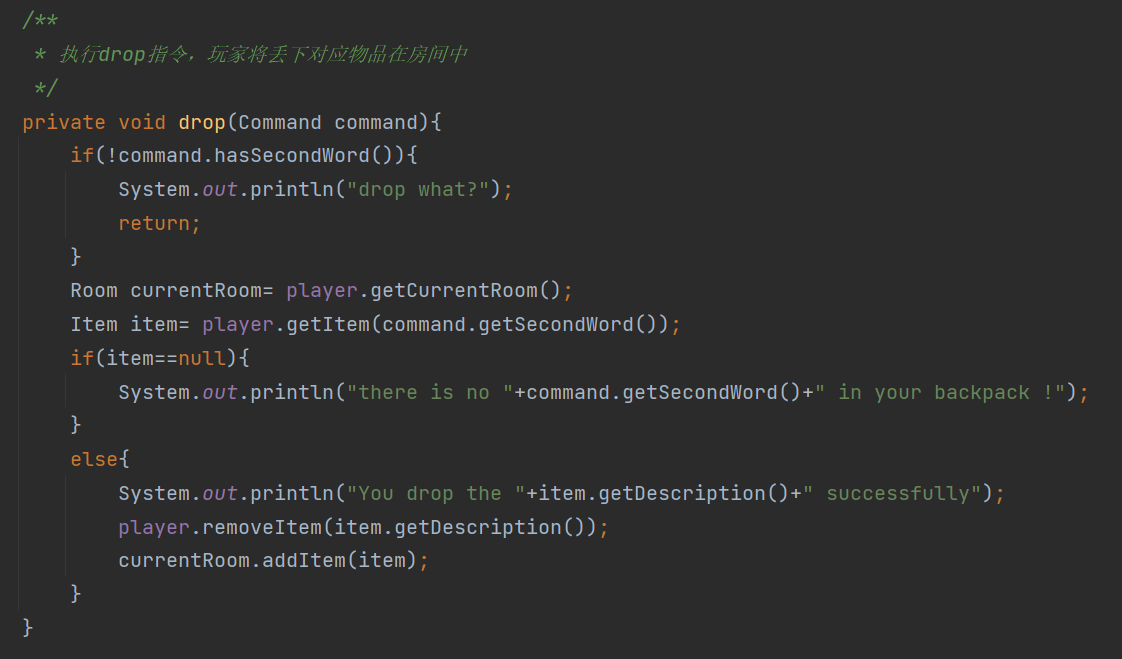
6、take指令的实现



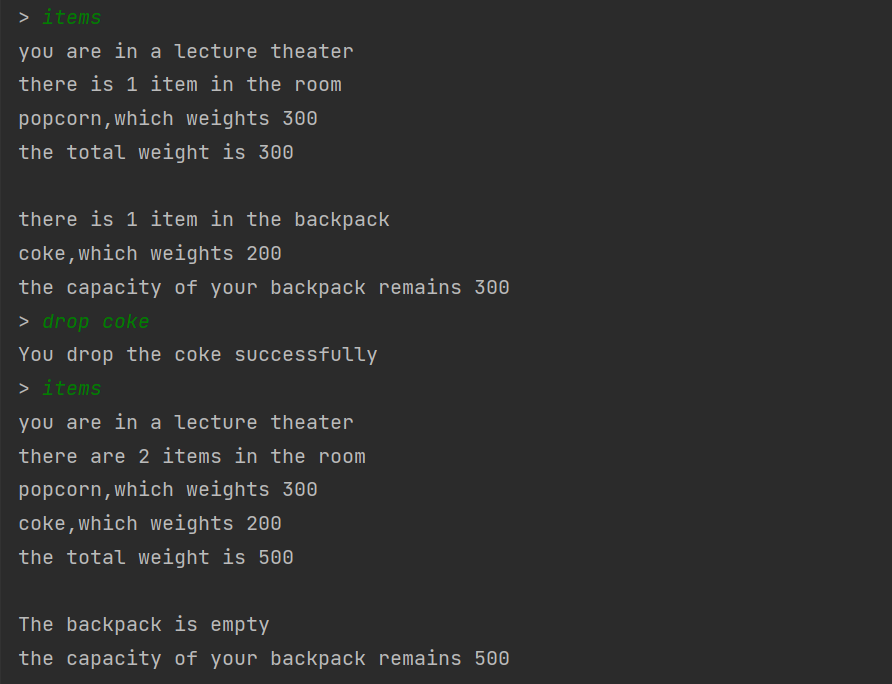
效果演示：



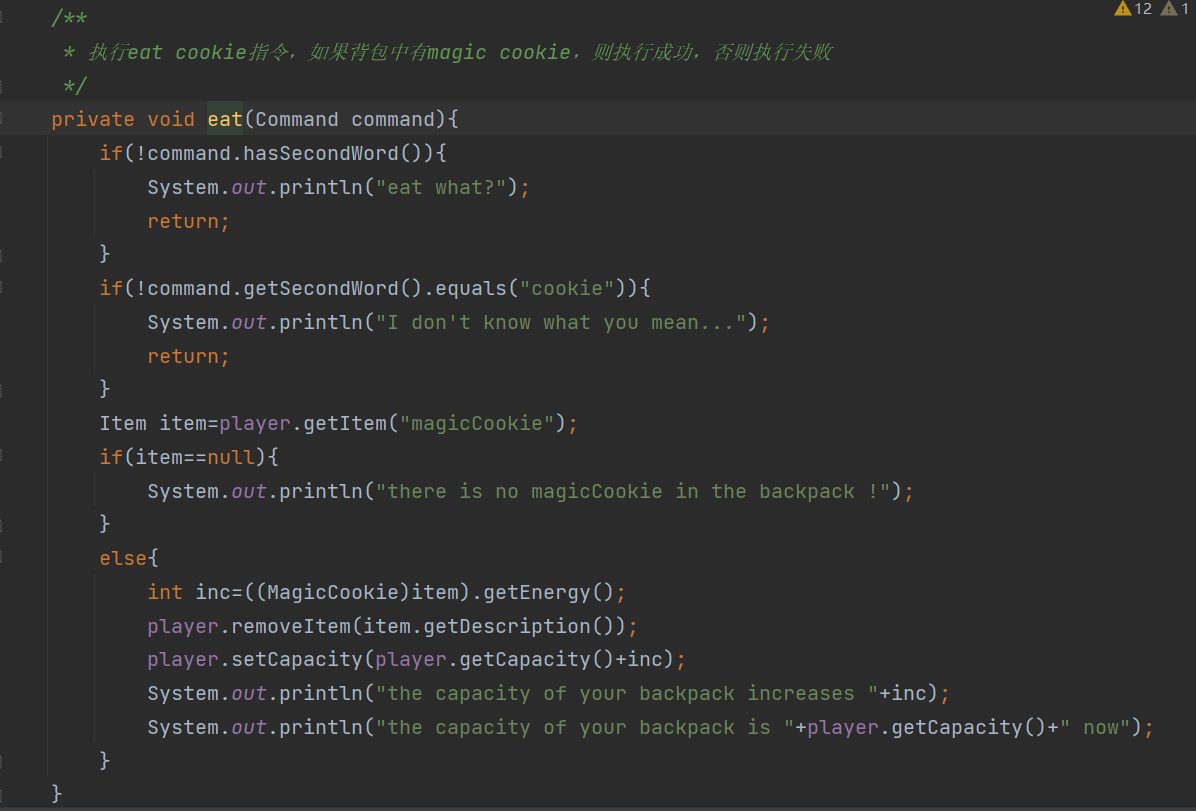
7、drop指令的实现



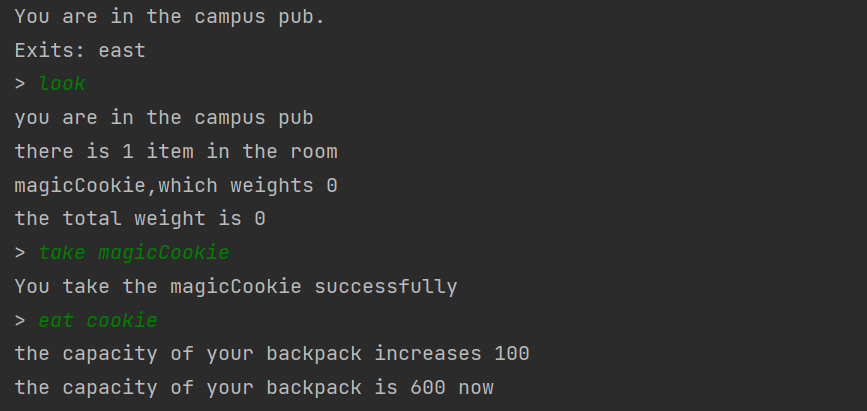
效果演示：



8、eat cookie指令的实现



效果演示：



**5、编写测试用例**

地图： pub-----outside-----threater

|

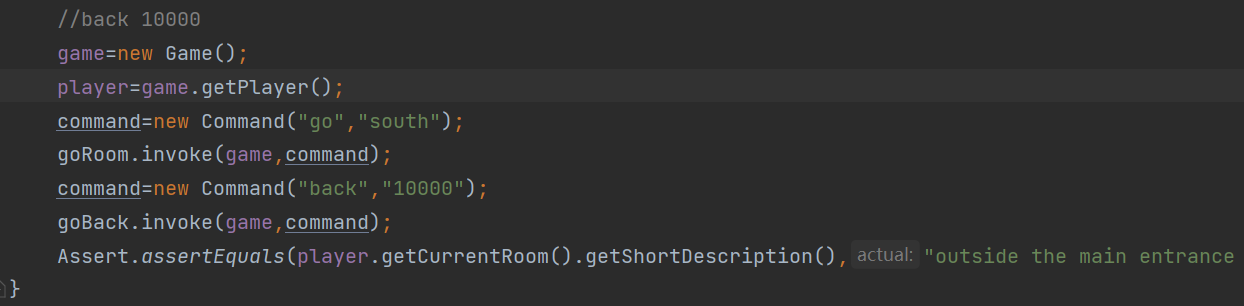
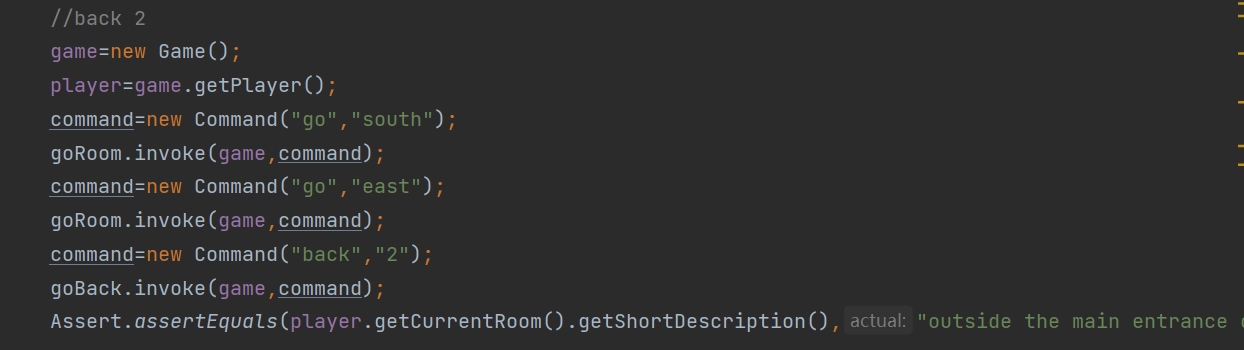
Lab -----office

|

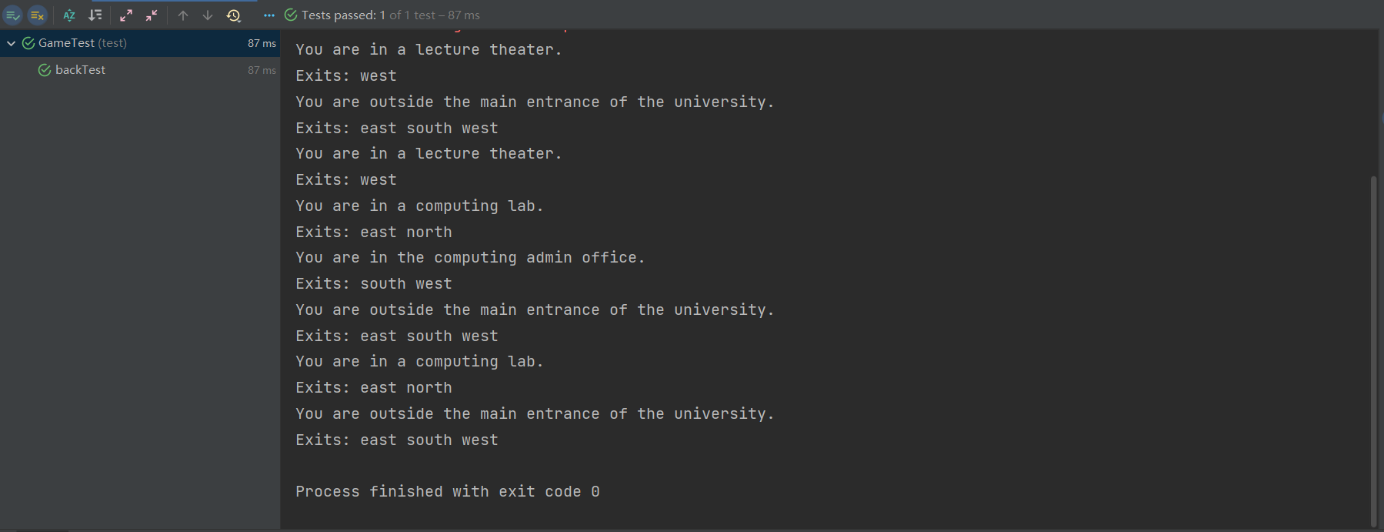
transferRoom1

本次测试用例主要编写的是back，take，drop，eat cookie指令的测试用例。

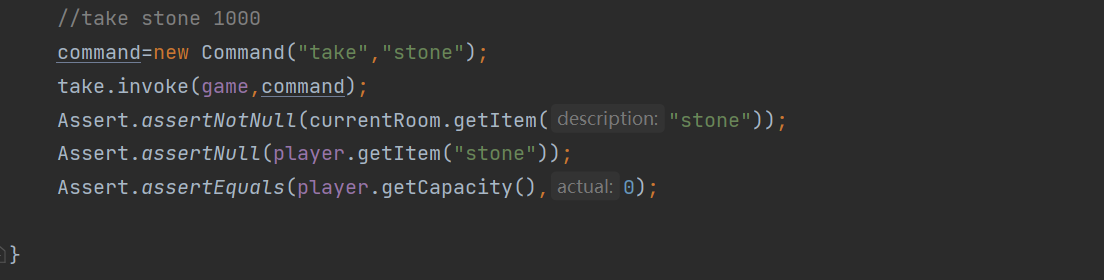
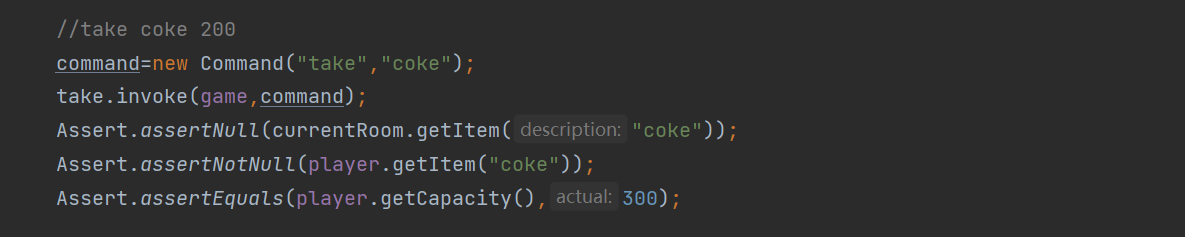
1、back指令的测试



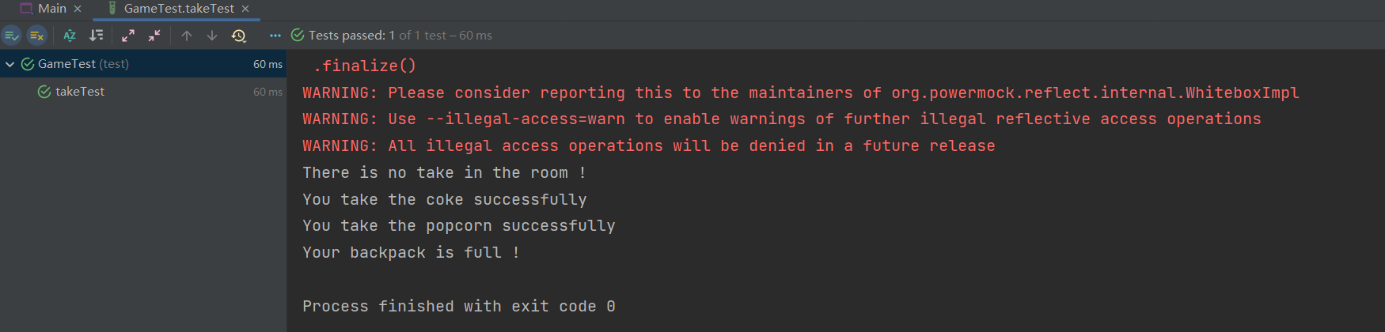
测试结果：



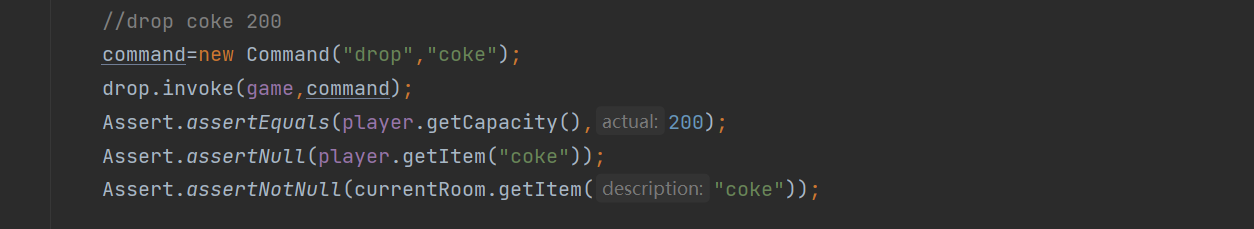
2、take指令的测试



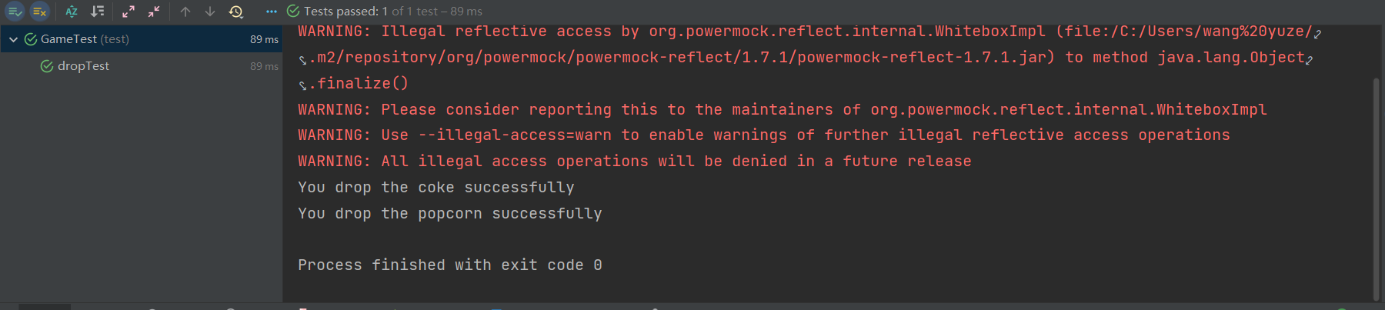
测试结果：



3、drop指令测试方法的编写

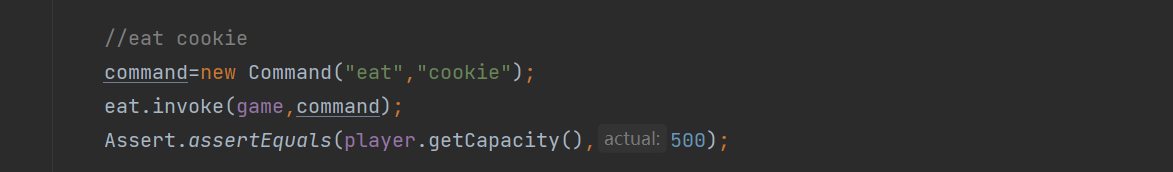


测试结果：



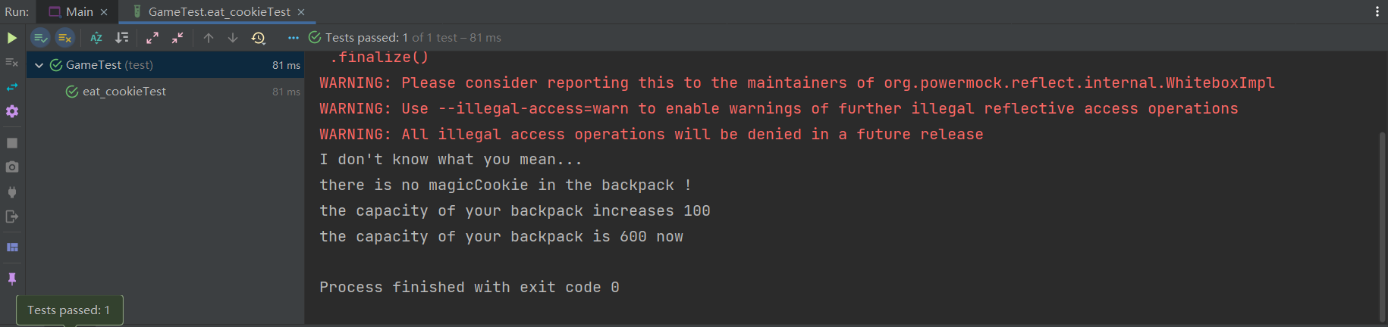
4、eat cookie的测试的编写







测试结果：



# 7 实施过程问题记录与分析

Junit测试中遇到的问题

java.lang.NoSuchMethodError:org.junit.runner.Description.createSuiteDescription(Ljava/lang/String;[Ljava/lang/annot

原因：引入junit的版本太低

Junit无法直接测试private方法，需要使用powermock利用java的反射机制进行测试。

Java.lang.NullPointException

原因：使用hashmap储存的对象，而对应key值输错了，得到value=null，所以在进行对item对象进行操作时，要现判空值，再进行后续操作。

# 8 任务总结

本次实验是一次个人开发的实践，收获颇丰，学习掌握了markdown语法进行书写，同时学会了git的基本功能，同时使用了github进行版本管理，对world of zuul

游戏也进行了一些功能上的补充和扩展，学习使用了powermock框架来进行私有方法的测试。这是一次综合性较高，难度较大的软件开发实践，从最开始的措手无策，通过网上查阅资料学习相关知识，慢慢地对项目开发有了自己的看法，最后独立完成了这次实践任务，成就感满满，不过其中还是有一些不足，等将来能力更进一步后再进行完善吧，这次实践，明白了一些基本的开发工具的使用，为以后进行软件开发打下了基础。

# 9 参考文献

# [1]软件工程中的UML建模技术,陈冠元，电子技术与软件工程. 2018,(05)

# [2] JUnit在单元测试中的应用，贾美丽，康珺，计算机与现代化. 2013,(08)

**《软件工程实践（一）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | | **王煜泽** | **学 号** | **0122010870108** | |
| **专业、班级** | | **软件工程 软件2002班** | | | |
| **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：**

**2023 年 1 月 3 日**