|  |  |
| --- | --- |
| **学 号：** | **0122010870414** |

****

**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** | **曾梓迪** |
| **编 号** | **0122010870414** |
| **指导教师** | **唐祖锴** |

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

实践的主要内容是: 在基于GitHub的分布式协同软件开发平台, 以个人开发方式对样例代码进行阅读、分析、标注和扩展, 并进行单元测试。

实践的任务目的是: 通过实践理解、掌握和应用软件工程的思路、方法、技术和工具, 从实践任务中培养个人软件工程能力。还要理解面向对象程序设计方法在实践工程项目中的应用,学会利用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。

# 2 任务分析

本次实践任务的工作重点:

对给定项目的源代码进行阅读,并理解其中的逻辑，然后在代码上进行规范的标注。并且要对给定的代码进行优化，寻求可行的优化方案；还需要对代码进行功能拓展，使之更加完善。最后还需要对代码进行单元测试。

本次实践任务的技术方案:

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；面向对象程序设计方法在实践工程项目中的应用；利用Git工具支持个人代码版本管理；利用CodeStyle等代码规范检查工具进行规范性约束和检查；基于Junit框架实施单元测试和测试驱动开发。

本次实践任务的工程实施方案:

完成软件系统的源码的阅读后，对源码进行批注，然后在此基础上做出优化，最后根据自己的需求拓展软件系统的功能后，进行单元测试。

本次实践任务的工具使用方案:

编译器: Jetbrain公司的idea

版本控制工: Git

包管理工: Maven

任务清单: Notion

浏览: Google Chrome

本次实践任务的重难点分析:

刚clone代码的时候对代码不熟悉，需要了解代码相关背景之后根据代码注释静心阅读代码;对代码进行功能拓展的时候需要有想象力，同时也需要能够将想象力用代码实现，这是有难度的。

# 3 开发计划

项目开发计划: 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；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

阅读和理解样例代码阶段采用的软件工程工具: git、idea

标注样例工程中的代码阶段采用的软件工程工具: 查阅javadoc规范

改进维护样例工程阶段采用的软件工程工具: ESLint

扩充样例功能阶段采用的软件工程工具: Git、idea

编写测试用例阶段采用的软件工程工具: JUnit4

项目遵循的主要管理策略和方法: 明确项目的设计方案，合理安排实施计划，确定项目需求;配置好完善的开发环境、工具及资料，使项目可以有效实施; 实施项目测试与审查，确保项目按计划进行。

# 4 软件配置计划

1、编码规范：

（1）源代码文件及版本应遵循UTF-8编码，避免出现其他中文及英文字符编码；

（2）每次迭代都要做好模块编码，避免后期模块出现冲突；

2、命名规范：

（1）源文件及文件夹命名应为全小写，词间用“\_”链接；

（2）变量命名应为驼峰式，第一个单词首字母小写；

3、分支管理规范：

（1）每次新增功能开发时在主分支上创建新的分支，分支命名遵循规范；

（2）新增功能完成后，按照分支管理机制进行合并；

4、提交规范：

（1）每次提交都要添加提交信息，描述本次提交的内容及修改理由；

（2）每次提交都要按照模块进行拆分，尽量减少提交代码量；

# 5 测试计划

规划: 主要测试规范包括：测试用例编写、测试环境准备、测试数据准备、执行测试用例、记录测试结果和总结测试结论等流程，要求每个单元都要进行测试，关键功能的每一步都要进行测试，以提高质量。可以从项目入口开始，也就是Main类的main方法开始加@Test注解进行测试,然后再对各个模块进行单元测试。

测试工具: Junit4。它是一个类包，提供了各种方法来测试Java类中的方法(method)。它有简单的注解，供书写Junit测试的基本特性断言方法(Assert Methods)比较测试的方法执行结果值和期望值@Ignore 注解，忽略测试方法或者测试类的执行期望异常测试超时测试 , 测试方法的执行时间测试组件，一起运行一些测试类参数化测试， 以不同的输入参数值测试方法。

主要测试规范:

1.测试计划

对于每个项目，应该制定一个明确的测试计划，可以根据需要进行调整，包括测试的目的、实施的工具和时间等，以及测试的具体细节，确保测试的过程严格按照计划实施。

2.测试策略

根据项目的具体内容，确定测试策略，实施测试。

3.测试环境

在测试前，必须要先搭建一个完备的测试环境，包括项目的搭建、数据的准备等，确保可以正常地实施测试工作。

4.测试标准

在实施测试时，必须确定合适的测试标准，然后按照标准进行操作，确保测试结果的准确性和可靠性。

5.测试方法

根据实际需要，可以使用不同的测试方法，比如单元测试、系统测试等，以确保测试质量和效率。

# 6 实施情况

功能设计方案:

1. "Replace Conditional with Polymorphism": 利用多态来消除多重分支,并使得命令的处理更加模块化。解决了一个问题:” 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。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。”
2. 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。
3. 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
4. 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

完成情况:

全部完成。其中"Replace Conditional with Polymorphism"这个优化给我后面拓展项目功能提供了极大的方便，方便了我在项目中拓展命令。

测试情况:

对项目进行了单元测试，没有出现问题。

软件代码提交情况:

已经将各模块相关代码全部提交。

# 7 实施过程问题记录与分析

问题记录:

对于有关传送功能房间的设计。该房间应该如何设计，才能使其具有不同于其他普通房间的功能，比如说传送。

分析:

房间就是Room类，普通房间有着一切Room类的方法。而现在有一个特殊的房间，有着特殊的方法与属性，但最关键的是它同样需要有所有普通房间的方法与属性，由此我便想到了继承:定义一个MagicRoom类，让它去继承(extends)Room类，由此便可在Room类的基础上为MagicRoom类拓展特殊功能，使其不同于普通房间。

# 8 任务总结

项目实践完成的总体情况: 对样例代码已经完全理解，并写出了规范的批注，同时也在样例代码的基础上进行了部分优化与功能扩展。

经验: 可以利用面向对象中的多态来消除if、else多重分支，既能使项目更加模块化，又能避免代码随着功能的拓展变得膨胀与臃肿。一句话概括就是"Replace Conditional with Polymorphism"。

收获: 这次项目实践让我对面向对象的理解加深了很多，以及看到了这种面向对象的思想是如何应用到实际项目中的。正因为利用了面向对象的思想，该项目源码的可读性、优化性与功能的可读性有了大大提高，让我明白了自己以后同样可以利用面向对象的思想来构建、设计一个项目；学会了markdown语法和利用mermaid插件生成UML类图。

# 9 参考文献

1. 《软件工程》(Software Engineering)，ISBN：978-7-5678-6839-5，作者：陈新，朝华出版社，2011年。
2. 《软件工程：原理、模式和工具》(Software Engineering: Principles, Patterns and Practices)，ISBN：978-0-321-45892-2，作者：艾伦·穆勒，韦伯出版社，2009年。
3. 关于"Replace Conditional with Polymorphism":

<https://refactoring.com/catalog/replaceConditionalWithPolymorphism.html>

<https://juejin.cn/post/6844904008377696269>

1. 关于mermaid:

<https://hellowac.github.io/2021/12/21/mermaid%E6%96%87%E6%A1%A3%E4%B8%AD%E6%96%87%E7%89%88-%E8%AF%AD%E6%B3%95-%E7%B1%BB%E5%9B%BE.html#%E8%AF%AD%E6%B3%95>

**《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**