工作日志

|  |  |  |
| --- | --- | --- |
| 时间 | 工作内容 | 备注 |
| 2022年6月2日 | 实验内容：  1.通过实例，熟悉各种UML图的使用。参考教材P226，补充材料6-4皇家服务站的例子，学习分析其各种UML图的设计过程。参照上面的例子，回顾自己的项目曾设计过的UML图，分工协作，修改。  2.论述利斯科夫替换原则（里氏代换原则）、单一职责原则、开闭原则、德（迪）米特法则、依赖倒转原则、合成复用原则，结合自己的实践项目举例说明如何应用（保存到每个小组选定的协作开发平台上，以组为单位）  3.上网查询“阿里编程规范（如：阿里巴巴JAVA开发手册）；华为编程军规”等，对照自己的代码看有哪些不符合规范的地方，修改。  4.阅读下面软件测试相关资料（或查阅其它相关资料），了解软件测试的基本概念、主要技术、重要挑战等  组长陈文盼对工作进行了分配，郑晓旭负责总结UML图，我和史子涵同学负责说明各面向对象设计原则，赵一帆负责查看代码规范，陈文盼负责总结软件测试，并对实验报告作出总结。 | 实验内容和分工 |
| 2022年6月3日 | 我们学习和讨论了关于UML和面向对象设计原则、代码规范、软件测试等内容，我主要负责面向对象设计原则的部分内容。 | 学习讨论 |
| 2022年6月5日 | 总结了部分面向对象设计原则：  **开闭原则**  在面向对象编程领域中，规定“软件中的对象（类，模块，函数等等）应该对于扩展是开放的，但是对于修改是封闭的”，这意味着一个实体是允许在不改变它的源代码的前提下变更它的行为。该特性在产品化的环境中是特别有价值的，在这种环境中，改变源代码需要代码审查，单元测试以及诸如此类的用以确保产品使用质量的过程。遵循这种原则的代码在扩展时并不发生改变，因此无需上述的过程。  **依赖倒转原则**  依赖倒置原则的原始定义为：高层模块不应该依赖低层模块，两者都应该依赖其抽象；抽象不应该依赖细节，细节应该依赖抽象（High level modules shouldnot depend upon low level modules.Both should depend upon abstractions.Abstractions should not depend upon details. Details should depend upon abstractions）。其核心思想是：要面向接口编程，不要面向实现编程。  依赖倒置原则的目的是通过要面向接口的编程来降低类间的耦合性，所以我们在实际编程中需要遵循以下4点：每个类尽量提供接口或抽象类，或者两者都具备；变量的声明类型尽量是接口或者是抽象类；任何类都不应该从具体类派生；使用继承时尽量遵循里氏替换原则。  **合成复用原则**  合成复用原则（Composite Reuse Principle，CRP）又叫组合/聚合复用原则（Composition/Aggregate Reuse Principle，CARP）。它要求在软件复用时，要尽量先使用组合或者聚合等关联关系来实现，其次才考虑使用继承关系来实现。合成复用原则是通过将已有的对象纳入新对象中，作为新对象的成员对象来实现的，新对象可以调用已有对象的功能，从而达到复用。  如果要使用继承关系，则必须严格遵循里氏替换原则。合成复用原则同里氏替换原则相辅相成的，两者都是开闭原则的具体实现规范。 |  |
| 2022年6月7日 | 作为本周工作的总结，我们建立了项目跟踪表，记录了每个人工作量、里程碑、工作进度，以便于记录各人的工作完成情况，之后的每周都将依据实际情况对其进行更新。我也完成了自己的工作日志。 | 工作总结 |