工作日志

|  |  |  |
| --- | --- | --- |
| 时间 | 工作内容 | 关键词 |
| 2022-5-26  星期四 | 小组召开电话会议商议任务分配。每个人负责不同的模块，我所负责的模块是学生选课模块。每个人都要结合项目的进程和开发历程，从设计原则的几个方面，组员对负责设计的模块进行评估，思考存在的问题和解决方案。 | 学生选课模块  设计方法  设计原则  面向对象 |
| 2022-5-27  星期五 | 今天我主要做了实验十一的任务四，主要内容如下：  采用了面向对象的设计方法。  学生选课信息管理系统基于数据库进行操作，有学生、管理员、课程信息、用户信息等多种对象实体，不同的实体有不同类型的数据，所有的操作都是针对这些对象展开的。 | 设计方法  面向对象 |
| 2022-5-28  星期六 | 今天我主要做了实验十一的任务二，我对我负责的学生选课这一模块根据设计原则进行了评估。主要内容如下：  负责设计的模块：学生选课模块。学生端可以通过登录该模块，选择课程进行选课。  问题1：如果一个学生在查看课程信息，刚好某个管理员登录系统修改了课程信息，那么学生是否会看到错误的课程信息？  解决方案：可以设置专门的时间段，时间段内由管理员对课程信息进行维护，学生不可查看；该时间段外学生可自由登陆。  问题2：如果两个学生同时选课，课容量信息如何更改？  解决方案：使用锁来控制每次只有一个学生选课访问并更改课容量信息。 | 学生选课模块  根据设计原则进行评估 |
| 2022-5-29  星期日 | 我分到了本周任务的任务三：学习依赖注入技术，主要内容如下：  依赖注入是解决一个对象如何获得它依赖的其他对象的技术。例如，一个对象A可以通过注入器(Injector)获得它所依赖的其他对象B、C。其中B、C对象是A对象的依赖(Dependencies)被传入到A对象中，一般被称为Service，A作为接收对象一般叫Client，而负责将依赖传递入接收对象的代码叫做Injector。  因此我们可以看到Client具体使用什么Service是由Injector去指定的，并传入到Client中，而不是Client自己去指定或创建的，这一点也是依赖注入的基本要求。  依赖注入是控制反转(Inversion of Control)这种技术的一种形式。目的是为了进一步解耦对象。Client需要某种服务，但是它仅仅知道这种服务(接口），而不知道具体实现服务的是哪个类。Client把找到这个实现类的工作交给Injector去完成。Injector会把找到的实现类的实例通过Client提供的注入方式注入到Client中。这样一来Client就和具体的服务实现解耦了，Client既无需要知道具体的服务实现，也能使用到服务了。  依赖注入将Client的依赖的创建和Client本般的行为分割开来。依赖注入促进了松耦合编程、依赖反转、单一责任原则等这些编程设计思想。  依赖注入这种技术中，一般包含四种角色，各司其职：  1.Service 对象，它包含了Client要使用的服务。  2.interface接口，通过它Client只知道有什么服务，而不需要知道具体实现。  3.Client对象，它的行为依赖于它所使用的服务  4.Injector注入器，它负责构造具体的Service并将其传入到Client对象中。  Client通过以下方式(不限于)来向Injector提供注入口：  1.构造器注入，即通过Client的构造器注入，简单来说就是new这个Client对象时，通过它的构造函数把依赖传进去。  2.set方法注入，即Client对象提供一些set方法，Injector就可以调用set方法把依赖传进去。  3.接口注入，即这些依赖的接口提供一个注入器方法 | 依赖注入 |
| 2022-5-30  星期一 | 小组讨论每个人所负责的任务，其中关于设计方法方面大家有些分歧，有的认为应该使用自顶向下的设计，有的认为应采用伪合理的设计过程，先设计最难的最不明确的部分。关于设计原则方面，每个人都说了自己所负责模块可能遇到的问题以及自己所能想到的解决方法。每个人都解释了一下自己所负责的模块采用了那些面向对象的设计。  本周由我负责汇总实验成果并总结报告。 | 成果汇总和讨论  报告撰写 |