第二次实验报告

杨天悦 2018302140018

陈宥伊 2018302100123

**任务分工**

杨天悦：

负责用户类装饰器模式的代码编写、课程服务类过滤器模式的代码编写、数据库的查询操作。

陈宥伊：

负责课程服务类装饰器模式的代码编写、数据库的创建与连接、数据库的增删改操作。

**1．用户类的装饰器模式（杨天悦负责）**

**思路**

由于Student类与Teacher类扩展了User类的功能，因此设计装饰器模式，用于扩展这些功能。

**大致代码流程**

1. 创建接口UserOp，定义用户行为LogIn()和LogOut()。
2. 创建抽象类User类和UserDec类，均继承自UserOp。
3. 创建Student类和Teacher类，继承自抽象类User；创建StudentDec类和TeacherDec类，继承自抽象类UserDec。
4. 在StudentDec类中，添加方法ApplyForRepre()，进行课代表申请操作；在TeacherDec类中，添加方法SetRepre()，进行课代表设置操作。
5. 用StudentDec和TeacherDec装饰User类的对象，以实现学生与老师的不同操作。

**应用该模式的优势**

装饰类和被装饰类可以独立发展，不会相互耦合，可以动态扩展User类的功能。

**2．课程管理类的过滤器模式（杨天悦负责）**

**思路**

由于课程列表中存在的课程较多，从数据库中查询耗时较高，因此设计过滤器模式，筛选出满足一定条件的课程，便于之后的查询。

**大致代码流程**

1. 创建抽象类Criteria类，定义方法MeetCriteria()。
2. 创建CriteriaCompulsory、CriteriaOptional、CriteriaPublic、CriteriaSpecialized四个类，继承自Criteria类，使用方法MeetCriteria()实现具体的过滤操作。
3. 在查询方法中，根据过滤标准，选择用不同的构造方法初始化Criteria类的对象，对课程列表进行过滤筛选。

**应用该模式的优势**

提高了查询操作的效率，不必在每次查询时，都要进行数据库的访问。

**3．课程服务类装饰器模式（陈宥伊负责）**

在对课程的操作方面，学生和老师两个实体类分别拥有不同的权限，所以为了更加灵活地为老师和学生提供不同的服务，我们选择用装饰器模式来实现这个功能。

首先我们先将对课程的操作封装成一个CourseService接口（增、删、改、查）。再分别创建StudentCS类和TeacherCS类来分别实现接口。（对于学生来说，增即为将一个未选课程，删即删除一个已选课程，没有修改课程权限，查即按照不同规则查询；对于老师来说，增即为新增一个未选课程，删即为删除一个课程，改即为修改课程，查即为按照不同规则查询）最后为了方便更加灵活地赋权给学生和老师，封装了一个CourseServiceDecorator类，继承于CourseService接口，内含一个protected CourseService courseService字段。这样只需要在实例化这个类的时候，赋值这个字段为StudentCS类或者TeacherCS类就可以实现不同的赋权了。

**4．本地数据库连接（陈宥伊负责）**

单独建立一个MysqlClient类，利用dataset连接本地mysql数据库。实现不同的函数以实现学生和老师不同的课程服务需求。