**设计模式学习与项目经验总结**

在这个学期，我跟随着侯老师，认真地学习了设计模式，并且通过做项目的方式亲自实现了一些设计模式，收获颇丰。现在这一期课程即将结束，我希望能对所学知识进行一个总结。

设计模式是针对软件开发过程中经常发生的问题的解决方案。设计模式可以提高代码的复用性、可移植性和可读性，当功能需要修改、拓展时，设计模式可以规避牵一发动全身的问题，从而大大提高软开效率。

设计模式分为创建型模式、结构性模式和行为模式。创建型模式用于初始化对象，可以进一步划分为使用继承机制的类创建模式和将初始化委托给另一个对象的对象创建模式，它们帮助一个系统独立于如何创建、组合和表示它的那些对象，可以避免硬编码的发生，从而赋予对象的创建以极大的灵活性；结构型模型涉及到如何组合类和对象以获得更大的结构，结构型模式采用继承机制来组合接口或实现，其并非是通过对接口函数进行组合，而是通过对对象进行组合，来实现新功能；行为模式涉及到算法和职责的分配，行为模式不仅描述对象或类的模式，还描述它们之间如何进行互动，行为对象模式使用对象复合，使得对象可以相互协作以完成其中任何一个对象都无法单独完成的任务。

在项目中，我通过实践深入理解了其中的三个设计模式——组成(Composite)、装饰(Decorator)与策略(Strategy)模式，并在和同学写的其他设计模式耦合的过程中深入理解了更多的设计模式。

Composite设计模式是一种结构型模式，将对象组合成树形结构以表示“部分-整体”的层次结构，好处是能够统一地处理单个和整体的对象。在我们组选择的奥林匹克项目中，需要展示所有项目的信息，而所有项目具有一定的层级。比如所有的奥运项目包括跑步、游泳、羽毛球等项目，而跑步、游泳中又包含更小的项目，比如男子100m游泳，女子1000m赛跑等等。项目可以分为复合项目和原子项目，要统一地对复合项目和原子项目进行增加、删除、展示等操作，并体现层级关系，Composite设计模式是最好的选择。而在创建Composite对象的时候，我发现可以使用Builder这种创建型设计模式，其将一个复杂对象的构建与它的表示分离开来，使得同样的构建过程可以创建不同的对象。而Composite对象包括复合项目和原子项目，如果使用Builder，则可以使用同一个创建函数来创建它们。同时，由于需要对Composite中某个项目的子项目进行查看，我发现很适合使用Iterator这种行为设计模式，其提供一种方法顺序访问一个聚合对象中各个元素，而又不需暴露该对象的内部表示。于是我和写Builder和Iterator设计模式的同学合作，共同完成了这个功能模块。

Decorator设计模式也是一种结构型模式，能够动态地给一个对象添加一些额外的功能。在奥运会项目中，需要实时地展示一些比赛过程，而比赛场景可以分成基本信息的打印（比赛名字，时间，场地，选手名称、国籍等）、以及该场比赛特有的信息——比如乒乓球比赛需要打印双方选手的比分；跑步需要记录时间和各个选手的排名。使用decorator可以很方便地把基本信息和扩展信息组合起来展示。而比赛的基本信息和拓展信息需要储存在对象中，由于基本信息是所有比赛所共有的，可以抽象出来，而拓展信息是不同比赛特有的，则需要在子类中给出。我发现可以使用AbstractFactory很方便地产生比赛信息类，AbstractFactory提供一个创建一系列相关或相互依赖对象的接口，而底下的子类负责创建对象。使用AbstractFactory，而不是FactoryMethod，就可以使用一个统一的创建函数，通过传入抽象类的不同子类作为参数，来创建不同的比赛信息类。这样，我与写AbstractFactory的同学进行合作，完成了这个功能模块。

Strategy是一种行为型设计模式，其意图是定义一系列的算法，把它们一个个封装起来，并且使它们可相互替换。本模式使得算法可独立于使用它的客户而变化，在需要增加或者修改算法时有着极大的便利。在本项目中，需要用不同的方式展示场地信息——例如通过场地名字首字母排序，通过场地编号顺序，通过在场地上举办活动的时间顺序来展示场地。未来有可能修改展示方式，或者增加新的展示方式，而使用Strategy设计模式，则只需要改变展示类的子类，不需要改动程序的其他部分，就可以做到这些事情，避免了未来可能出现的牵一发而动全身的窘境。

在学习设计模式的过程中，我复习了C++中虚函数和多态的知识，补全了一些以前未曾注意到的盲点。例如要用父类指针调用子类的函数，需要在父类中声明其虚函数；如果在父类中如果有纯虚函数，那么该父类则作为抽象类，不能创建对象，但是能建立指向父类的指针，而该父类的所有子类中需要具体化所有的纯虚函数，否则无法通过编译；不少设计模式通过将抽象类指针或者对象作为函数的参数，可以通过传入不同的具体类来实现不同的功能。

通过这个阶段的学习，我不仅学习了设计模式的知识，还通过与同学合作完成项目磨练了团队合作的能力。虽然这个学期的课程即将结束，但是设计模式的学习远远没有结束，我虽然对这23个基本的设计模式有所了解，但是对在实际项目中如何灵活地使用这些设计模式没有足够的经验，所以未来还要通过不断做项目，并在做项目时思考如何使用设计模式来融会贯通。