|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 设计模式名称 | 实现个（套）数 | 何处体现 | 备注说明 |
| 1 | ChainsOfResponsibility pattern | 1 | 动物运动会医院中，所有医生构成责任链 | 无 |
| 2 | Iterator pattern | 1 | 观赛前检票，检票员给观众检票就像遍历迭代器 | 无 |
| 3 | Specification pattern | 1 | 动物运动会最终评选奖项，通过一定的规约来决定每个动物的最终评奖等级 | 无 |

III.设计模式详述，按下列内容给出详细说明

3.1 设计模式ChainsOfResponsibility

3.1.1 实现API 描述

ChainsOfResponsibilityPattern类的成员函数run()具体实现了该模式。在该函数中，先通过创建一些医生并使他们构成责任链，然后接受病人请求，并返回给一个符合要求的医生来给其治病。

3.1.2 类图（还可添加流程图、时序图、等等其他UML图辅助说明）

图示

描述已自动生成

3.2 设计模式 Iterator

3.2.1 实现API 描述

IteratorPattern类的成员函数run()具体实现了该模式。在该函数中，通过访问观众audience类的迭代器来实现遍历整个观众队列。

3.2.2 类图（还可添加流程图、时序图、等等其他UML图辅助说明）

图示

描述已自动生成

3.3 设计模式 Specification

3.3.1 实现API 描述

SpecificationPattern类的成员函数run()具体实现了该模式。在该函数中，通过调用And()、Or()、Not()方法来实现不同规约之间的与、或、非逻辑组合。

3.3.2 类图（还可添加流程图、时序图、等等其他UML图辅助说明）

图示

描述已自动生成