实验四报告

# 策略模式

1. 应用场景分析

*描述飞机大战游戏中哪个应用场景需要用到此模式，目前代码实现中存在的问题及使用该模式的优势。*

策略模式（Strategy Pattern）是一种行为型设计模式，它能让你定义一系列算法，并将每种算法分别放入独立的类中，以使算法的对象能够相互替换。

本例中，代码直接调用shoot方法，无法使不同的射击方法分开，不利于后续添加弹道，原代码中（未删除，但本例实验中未调用）的shoot方法虽然实现了直射和散射的功能，但较为臃肿，采用策略模式能够使得代码模块化，同时不同策略的代码能够复用到不同类型的飞机上，提高了代码的复用性。

1. 解决方案

*将PlantUML插件绘制的类图截图到此处，并对UML类图中每个类、接口，以及其关键属性和方法进行简单说明。*

# 数据访问对象模式

1. 应用场景分析

*描述飞机大战游戏中哪个应用场景需要用到此模式，分析使用该模式的优势。*

数据访问对象模式（Data Access Object Pattern）用于把低级的数据访问 API 和操作从高级的业务服务中分离出来。本例中因为需要将排行榜每轮游戏结束时进行更新，需要使用此模式将数据的访问部分和游戏上层的结构分离出来，有利于模块化的管理和流程的控制。

本例中使用的DAO模式，采用json保存记录，记录的读取由DAO管理，使得类的功能相分离，提高了代码的复用性。

1. 解决方案

*将PlantUML插件绘制的类图截图到此处，并对UML类图中每个类、接口，以及其关键属性和方法进行简单说明。*