

（深圳）

实验报告

开课学期： 2024春季

课程名称： 面向对象的软件构造导论

实验名称： 飞机大战游戏系统的设计与实现

实验性质： 设计型

实验学时： 16 地点：T2210

学生班级： 计科7班

学生学号： 220110707

学生姓名： 苏圣迪

评阅教师：

报告成绩：

实验与创新实践教育中心制

2024年5月

# 实验环境

操作系统：windows

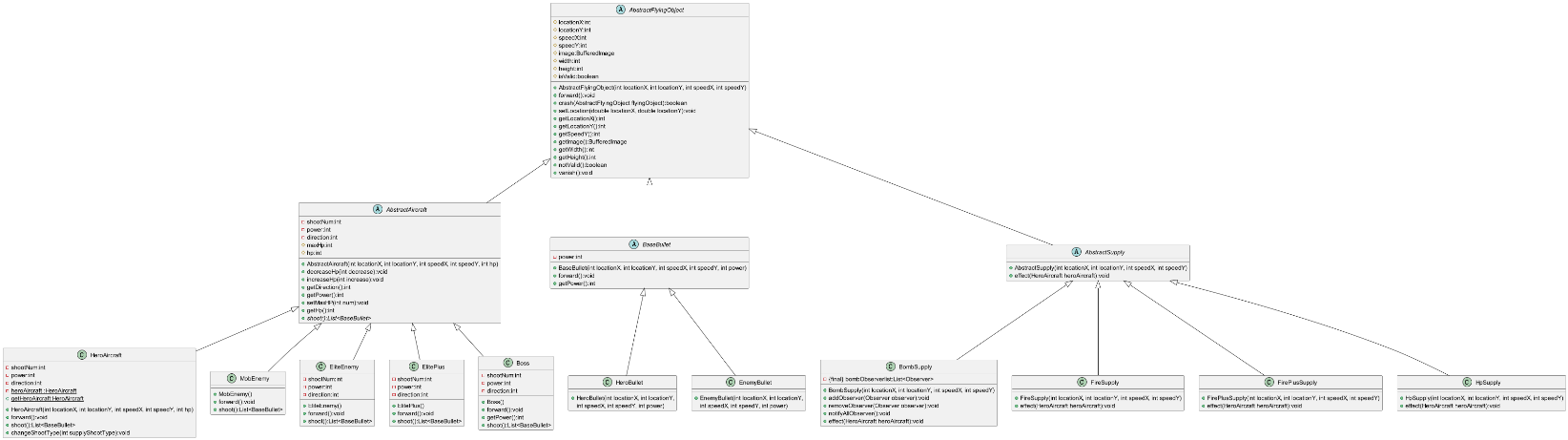
开发工具：Intellij IDEA\swing GUI\ JUnit

# 实验过程

以下UML结构图请更新为最终提交版本。

## 2.1类的继承关系

*请根据面向对象设计原则，分析和设计游戏中的所有飞机类、道具类和子弹类，并使用 PlantUML 插件绘制相应的 UML 类图及继承关系，类图中需包括英雄机、所有敌机、道具、子弹及它们所继承的父类。*



## 2.2设计模式应用

### 2.2.1单例模式

1. 应用场景分析

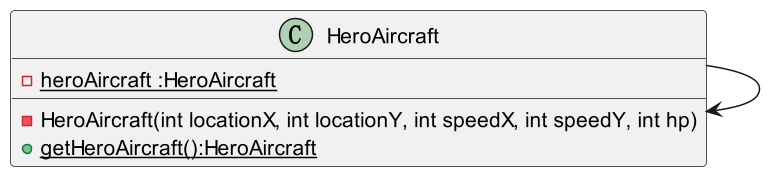
*描述飞机大战游戏中哪个应用场景需要用到此模式，设计中遇到的实际问题，使用该模式解决此问题的优势。*

采用单例模式创建英雄机。

单例模式可以节约系统资源，因为它确保一个类只有一个实例存在于内存中，避免了重复创建相同对象的开销。单例模式提供了一个全局访问点，可以在程序的任何地方访问该实例，使得数据共享变得简单。

1. 设计模式结构图

*结合飞机大战实例，绘制该场景下具体的解决方案（UML类图）。描述你设计的UML类图结构中每个角色的作用，并指出它的关键属性和方法。*



### 2.2.2工厂模式

1. 应用场景分析

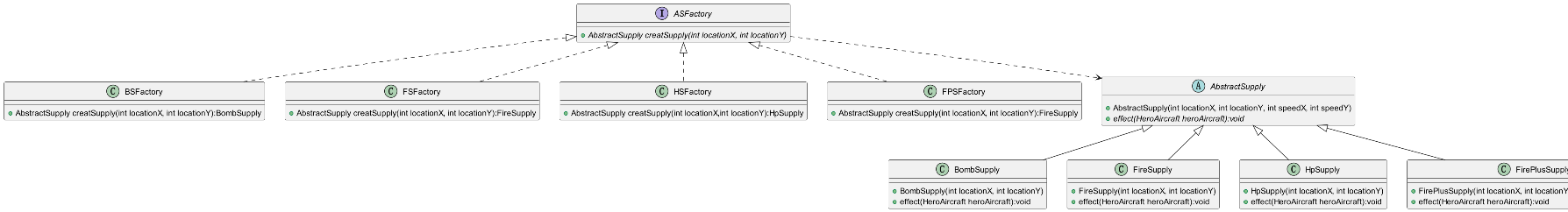
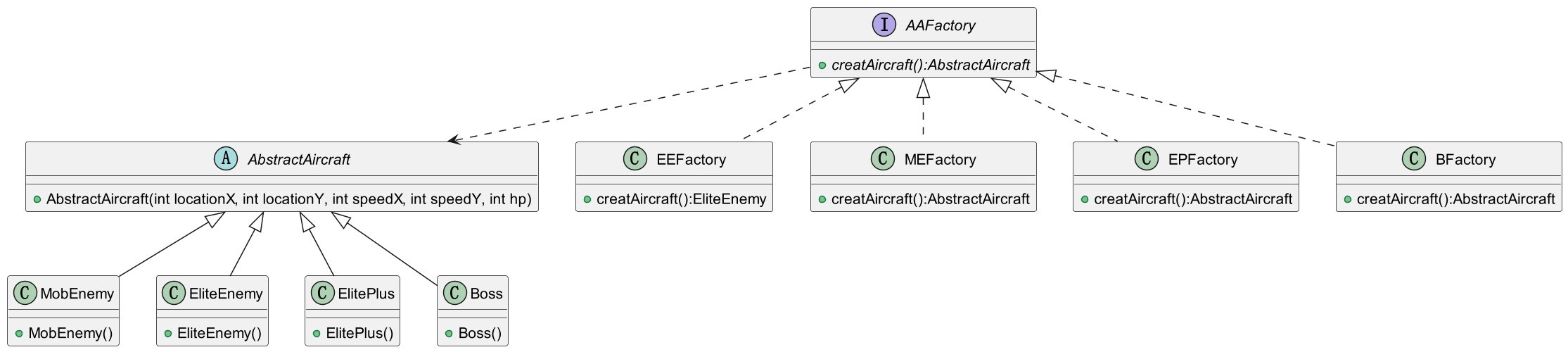
*描述飞机大战游戏中哪个应用场景需要用到此模式，设计中遇到的实际问题，使用该模式解决此问题的优势。*

采用工厂模式创建敌机以及四种道具。

工厂模式的优势包括：

解耦、封装、可扩展性、隐藏实现细节、代码复用设计模式结构图

*结合飞机大战实例，绘制该场景下具体的解决方案（UML类图）。描述你设计的UML类图结构中每个角色的作用，并指出它的关键属性和方法。*

### 2.2.3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，设计中遇到的实际问题，使用该模式解决此问题的优势。*

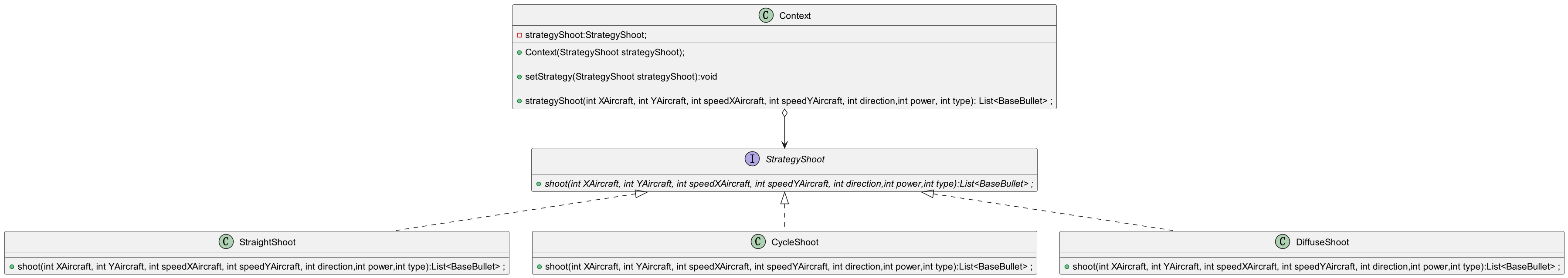
采用策略模式实现不同机型的弹道发射及两种 火力道具的加成效果

策略模式的优势：

灵活性和可扩展性、解耦、易于维护、单一职责原则

1. 设计模式结构图

*结合飞机大战实例，绘制该场景下具体的解决方案（UML类图）。描述你设计的UML类图结构中每个角色的作用，并指出它的关键属性和方法。*



### 2.2.4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，设计中遇到的实际问题，使用该模式解决此问题的优势。*

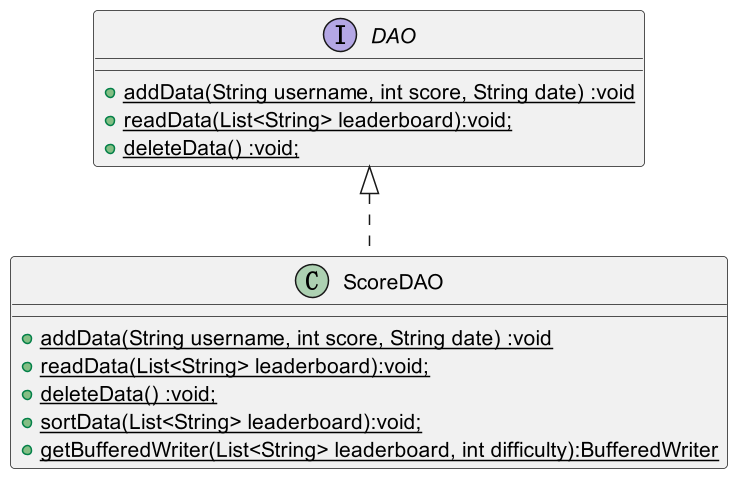
采用数据访问对象模式实现玩家的得分排行榜

数据访问对象模式优点：

分离数据持久化细节、提高代码可维护性、便于测试

1. 设计模式结构图

*结合飞机大战实例，绘制该场景下具体的解决方案（UML类图）。描述你设计的UML类图结构中每个角色的作用，并指出它的关键属性和方法。。*



### 2.2.5观察者模式

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，设计中遇到的实际问题，使用该模式解决此问题的优势。*

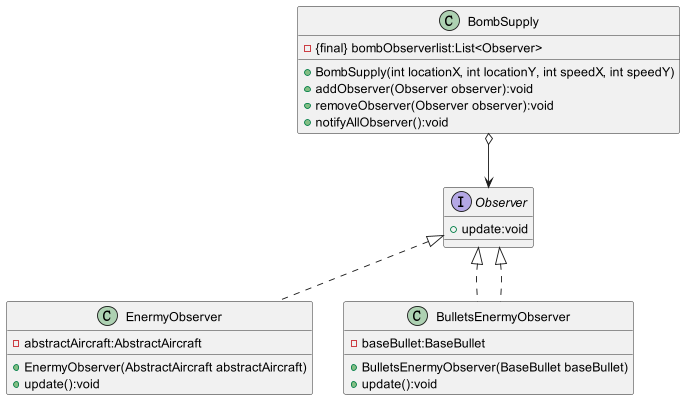
采用观察者模式实现炸弹道具

优势：

松耦合、易于扩展、通知机制

1. 设计模式结构图

*结合飞机大战实例，绘制该场景下具体的解决方案（UML类图）。描述你设计的UML类图结构中每个角色的作用，并指出它的关键属性和方法。*



### 2.2.6模板模式

1. 应用场景分析

*请简单描述你对三种游戏难度是如何设计的，影响游戏难度的因素有哪些。描述飞机大战游戏中哪个应用场景需要用到此模式，设计中遇到的实际问题，使用该模式解决此问题的优势。*

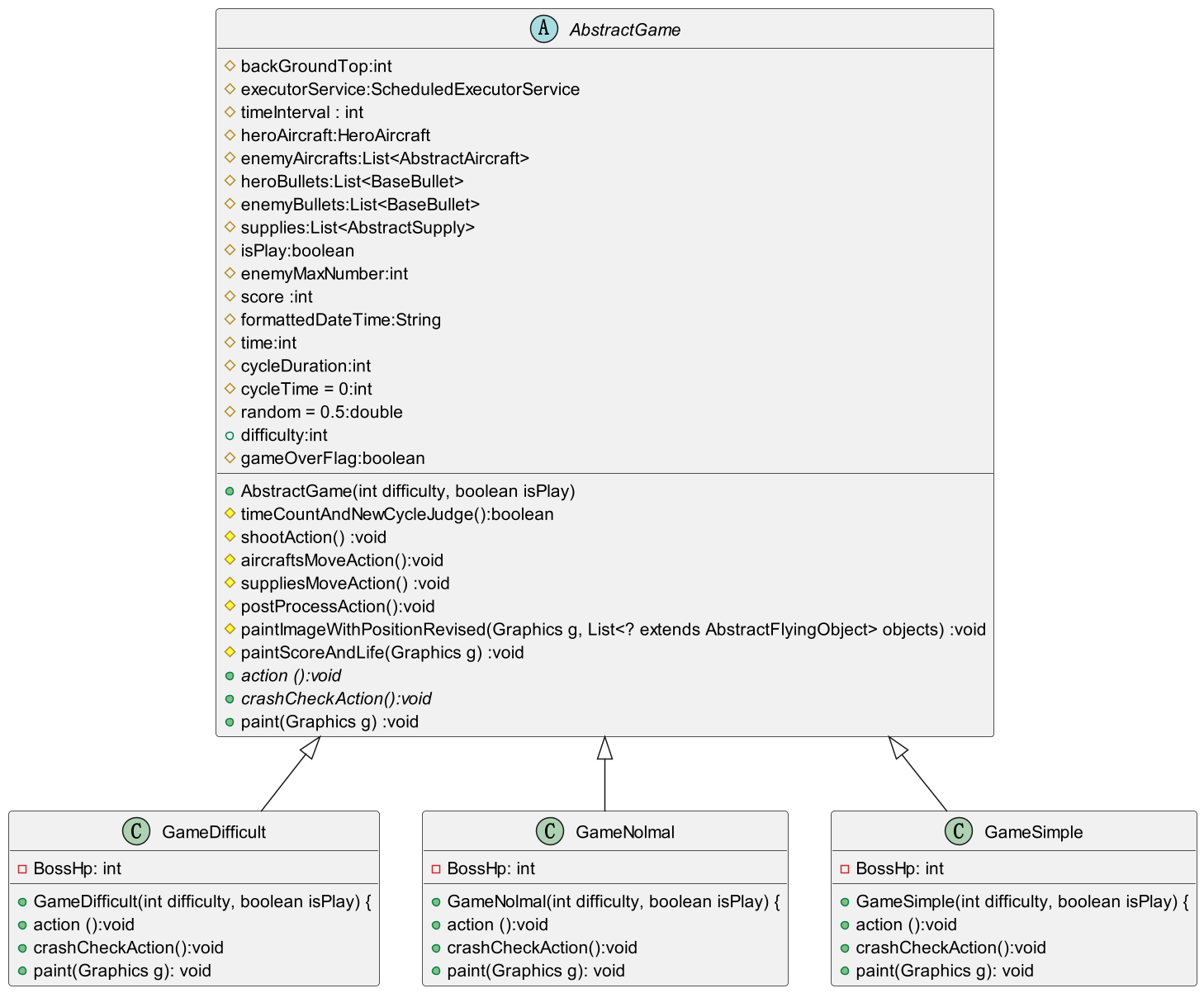
采用模板模式实现简单、普通、困难三种游戏难度

模板模式的优点:

代码复用、结构清晰、可扩展性、行为统一、符合开闭原则

1. 设计模式结构图

*结合飞机大战实例，绘制该场景下具体的解决方案（UML类图）。描述你设计的UML类图结构中每个角色的作用，并指出它的关键属性和方法。*



# 收获和反思

*请填写本次实验的收获，记录实验过程中出现的值得反思的问题及你的思考。欢迎为本课程实验提出宝贵意见！*

通过这次实验，我不仅学会了如何使用Java来开发游戏，还提高了自己的编程能力和解决问题的能力。对于课程实验，我建议增加一些与实际项目开发更相关的内容，例如团队协作、版本控制等，这样可以更好地帮助学生准备未来的工作。同时，提供更多的实践机会和项目案例也会让学生受益匪浅。