**《软件工程》**

**实验报告七：设计模式以及构件设计**

**姓 名： 胡国昌 学 号： 202210120518**

**院 系： 计算机与信息学院 专 业： 计算机科学与技术**

**实 验 室： J1-306 实验日期： 2024.12.02**

**总评成绩： 审阅教师： 杨青**

# 实验目的

1. 了解软件工程工具生成代码的方法，理解代码自动生成的过程，破除自动生成代码“黑科技”
2. 掌握构件设计方法
3. 学习Bridge设计模式，体会设计模式的作用

# 实验环境

Rational Rose Enterprise Edition, StarUML.

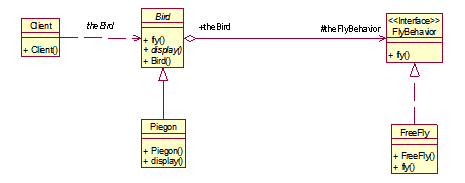
# 实验要求

1. 参考[第7章 设计概念（design principle &pattern）（4学时）](https://star.jmhui.com.cn/u/cms/www/202305/08213040s60e.pdf)，学习Bridge设计模式的使用；
2. 学习 Rose 生成代码的方法，理解代码自动生成原理
3. 文件以“学号-姓名-软件工程实验七.rar”的方式命名，把模型文件和工程实现代码打包，提交到长江雨课堂**软件工程实验七**；

# 实验内容

## 按照给出的 Bridge 设计模式 构建类图

有一个叫 HuntBird 的游戏，里面需要表示各种各样的鸟类。采用Bridge设计模式实现，针对各种鸟类，采用Bridge设计模式建模，如下图：



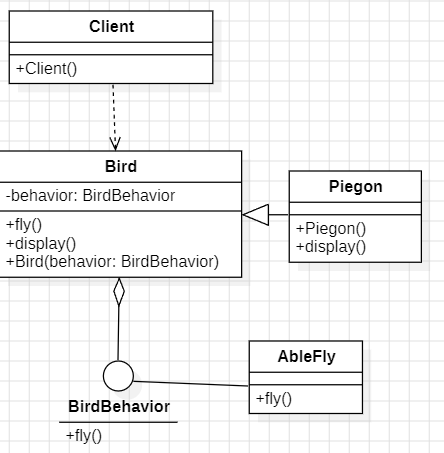
注意：

（1）绘制类之前，首先设置默认语言为Java/C++（本例以Java说明）

（2）Bird类为抽象类

（3）display方法为抽象方法

（4）Bird具有保护类型的FlyBehavior属性；



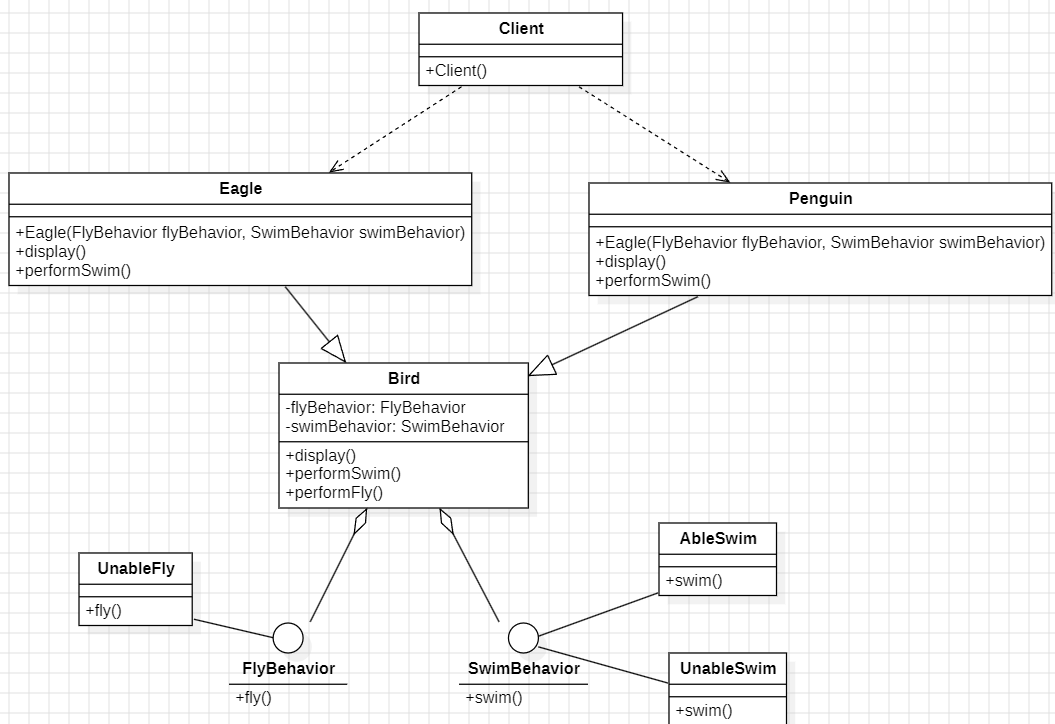
## 按照以下给出的要求，扩展类图，并生成代码框架，实现代码，测试

(1)添加一个飞翔的行为的实现类UnableFly，实现fly接口方法：扑腾几下飞不起来；

(2)添加游泳的行为SwimBehavior接口，两个实现类AbleSwim，实现swim接口方法：鱼翔浅底；UnableSwim实现类实现swim接口方法：扑腾几下就淹死了：

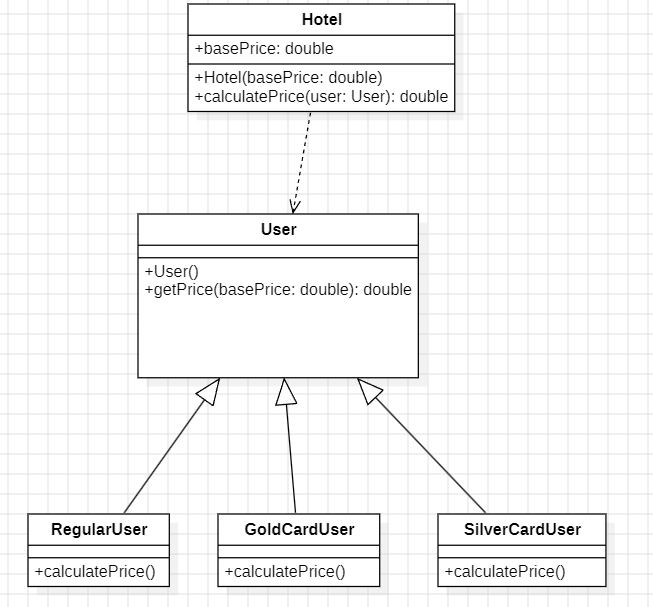
(3)添加两个具体类：老鹰Eagle和企鹅Penguin，并实现相应代码，即构造函数和display抽象方法

(4)使用Client类，测试以上代码



## 根据以下需求说明，设计符合面向对象设计原则的 UML 类图并实现代码。

假如我们正在开发一个酒店预订系统，针对不同的用户，我们需要计算出不同的房价。比如，普通用户是全价，金卡是 8 折，银卡是 9 折。



# 实验结论

通过本次实验，我们深入探讨了桥接（Bridge）设计模式在系统解耦和行为扩展中的重要作用，并成功完成了从类图设计到代码实现的全流程实践。实验的过程和结果表明，该模式能够显著提升系统的灵活性和可维护性。

首先，桥接设计模式的核心在于将抽象部分和实现部分分离，从而允许它们独立变化。在本实验中，我们通过 Bird 类的抽象设计以及 FlyBehavior 和 SwimBehavior 接口的实现，成功将鸟类的具体行为与其种类解耦。这种设计不仅减少了类之间的耦合度，还使得新增行为时无需修改已有代码，从而满足了开放封闭原则（OCP）。例如，通过新增 UnableFly 和 AbleSwim 类，我们仅需实现相关接口，无需改动 Bird 类及其子类的代码结构。

其次，实验充分体现了设计模式在代码扩展中的优势。通过新增 Eagle（老鹰）和 Penguin（企鹅）两个具体鸟类，我们验证了桥接模式的多维扩展能力。在传统设计中，可能需要为每种鸟类手动实现所有行为，但在本实验中，这些行为通过接口和实现类完成了复用，显著提高了开发效率。同时，行为的实现独立于具体鸟类，这种灵活性使系统能够快速响应需求的变化。

在测试阶段，我们通过 Client 类验证了系统设计的正确性和稳定性。具体测试包括实例化不同的鸟类并调用其特定行为，结果表明系统能够准确区分不同种类鸟类的行为表现。此外，我们还对新增行为（如 UnableSwim）的功能进行了测试，确保了代码的完整性与可靠性。

# 仓库地址

[https://github.com/xieyangyuyue /softProject.git](https://github.com/xieyangyuyue%20/softProject.git)