

重 庆 大 学

学 生 实 验 报 告

实验课程名称 软件架构与设计模式

开课实验室 DS1501

学 院 大数据与软件学院 年级 2021 级专业班软件工程 X 班

学 生 姓 名 XXX 学 号 2021XXXX

开 课 时 间 2023 至 2024 学年第 2 学期

总 成 绩	
教师签名	

大数据与软件学院制

《软件架构与设计模式》实验报告

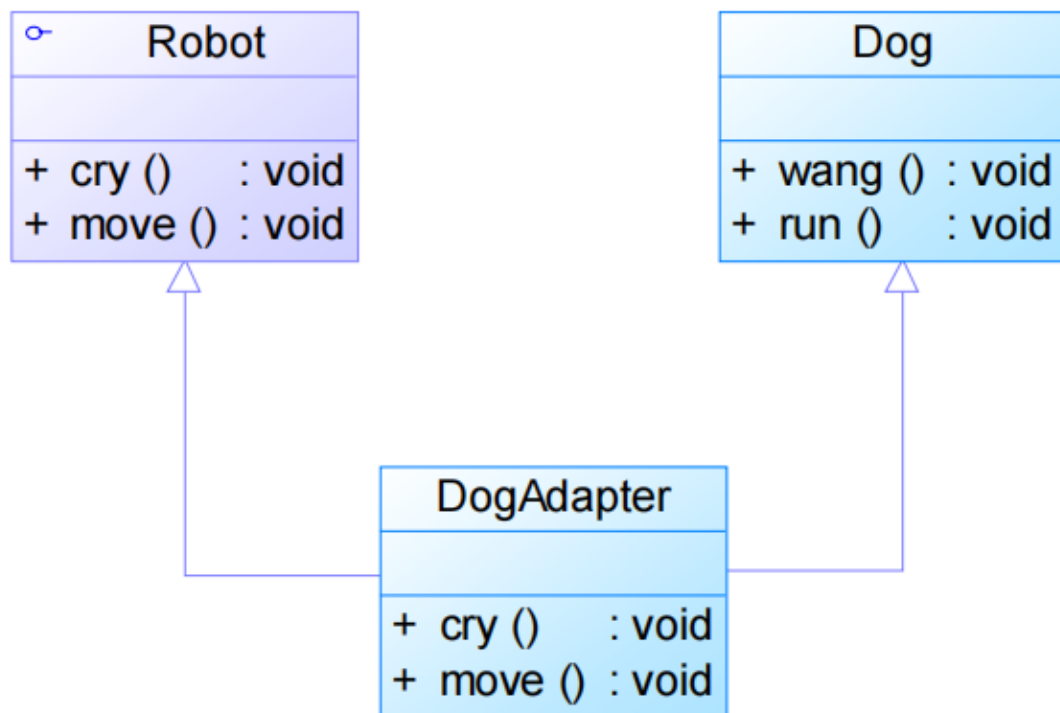
开课实验室：大数据与软件学院

2024 年 6 月 13 日

学院	大数据与软件学院	年级、专业、班	2021 级软件工程 X 班	姓名	XXX	成绩	
课程名称	软件架构与设计模式	实验项目名称	“结构型模式”的应用	指导教师	XXX		
教师评语	<div>教师签名：_____</div> <div>_____ 年 月 日</div>						
<div>一、实验目的 以设计适配器模式(Adapter)为实验实例，掌握“结构型模式”的工作原理、应用环境和应用方法。</div> <div>二、实验条件 计算机上安装 Word 字处理软件、Rational 软件、Eclipse 或其他编程环境。</div> <div>三、实验内容 用 Adapter 模式设计一个仿生机器人： 要求机器人可以模拟各种动物行为，在机器人中定义了一系列方法，如机器人叫喊方法 cry()、机器人移动方法 move() 等。如果希望在不修改已有代码的基础上使得机器人能够像狗一样汪汪叫，像狗一样快跑，或者像鸟一样叽叽叫，像鸟一样快快飞，使用适配器模式进行系统设计。</div>							

四、实验步骤

1、用 UML 设计“仿生机器人”的类图。



2、根据类图写出“仿生机器人”的源代码。

(1)Robot.java

```
// Robot interface
public interface Robot {
    1 个实现
    void cry();

    1 个实现
    void move();
}
```

(2)DogAdapter.java

```
// DogAdapter class
public class DogAdapter implements Robot {
    private Dog dog;

    // Constructor
    public DogAdapter(Dog dog) { this.dog = dog; }

    // Adapter method for cry
    @Override
    public void cry() { dog.wang(); }

    // Adapter method for move
    @Override
    public void move() { dog.run(); }
}
```

(3)Dog.java

```
// Dog class
public class Dog {
    // Dog's cry method
    public void wang() { System.out.println("Dog is barking: Wang! Wang!"); }

    // Dog's move method
    public void run() { System.out.println("Dog is running fast."); }
}
```

(4)Client.java

```
// Client code
public class Client {
    public static void main(String[] args) {
        // Create a Dog instance
        Dog dog = new Dog();
        // Create a DogAdapter instance
        Robot dogAdapter = new DogAdapter(dog);

        // Use DogAdapter to simulate Dog behavior as Robot behavior
        dogAdapter.cry(); // Expected Output: Dog is barking: Wang! Wang!
        dogAdapter.move(); // Expected Output: Dog is running fast.
    }
}
```

3、上机测试程序，写出运行结果。

结果正常输出且与预期相符。

```
Dog is barking: Wang! Wang!  
Dog is running fast.
```