

4. 编程题 (参考例子 13)

设计一个动物声音“模拟器”，希望模拟器可以模拟许多动物的叫声，要求如下。

- 编写抽象类 Animal

Animal 抽象类有两个抽象方法 cry() 和 getAnimalName(), 即要求各种具体的动物给出自己的叫声和种类名称。

- 编写模拟器类 Simulator

该类有一个 playSound(Animal animal) 方法, 该方法的参数是 Animal 类型。即参数 animal 可以调用 Animal 的子类重写的 cry() 方法播放具体动物的声音, 调用子类重写的 getAnimalName() 方法显示动物种类的名称。

- 编写 Animal 类的子类: Dog 和 Cat 类

图 5.18 是 Simulator、Animal、Dog、Cat 的 UML 图。

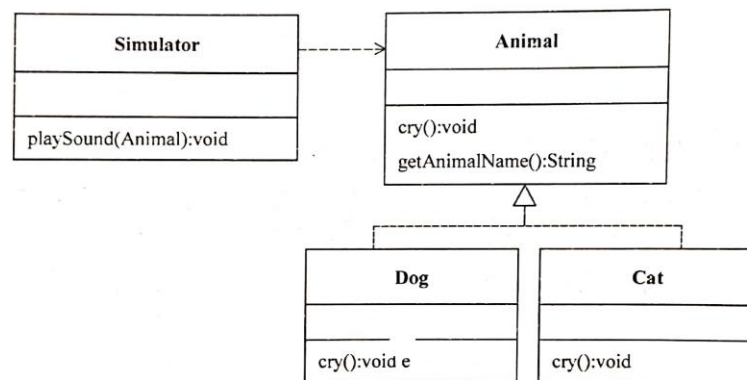


图 5.18 UML 类图

- 编写主类 Application (用户程序)

在主类 Application 的 main 方法中至少包含如下代码:

```
Simulator simulator = new Simulator();
simulator.playSound(new Dog());
simulator.playSound(new Cat());
```

例子 13

用类封装手机的基本属性和功能，要求手机既可以使用移动公司的 SIM 卡，也可以使用联通公司的 SIM 卡（可以使用任何公司提供的 SIM 卡）。

① 问题的分析

如果设计的手机类中用某个具体的公司的 SIM 卡，例如移动公司的，声明了对象，那么手机就缺少弹性，无法使用其他公司的 SIM 卡，因为一旦用户需要使用其他公司的 SIM 卡，就需要修改手机类的代码，例如增加用其他公司声明的成员变量。

如果每当用户有新的需求，就会导致修改类的某部分代码，那么就应当将这部分代码从该类中分割出去，使它和类中其他稳定的代码之间是松耦合关系（否则系统缺乏弹性，难以维护），即将每种可能的变化对应地交给抽象类的子类去完成。

② 设计抽象类

根据以上对问题的分析，首先设计一个抽象类 SIM，该抽象类有三个抽象方法：giveNumber()、setNumber()和 giveCorpName()，那么 SIM 的子类必须实现 giveNumber()、setNumber()和 giveCorpName()方法。

③ 设计手机类

设计 MobileTelephone 类（模拟手机），该类有一个 useSIM(SIM card)方法，该方法的参数是 SIM 类型。显然，参数 card 可以是抽象类 SIM 的任何一个子类对象的上转型对象，即参数 card 可以调用 SIM 的子类重写的 giveNumber()方法显示手机所使用的号码，调用子类重写的 giveCorpName()方法显示该号码所归属的公司。

例子 13 中除了主类外，还有 SIM 类及其子类：SIMOfChinaMobile（模拟移动公司提供的卡）、SIMOfChinaUnicom（模拟联通公司提供的卡）和 MobileTelephone 类。

图 5.16 是 MobileTelephone、SIM、SIMOfChinaMobile 和 SIMOfChinaUnicom 类的 UML 图，程序运行效果如图 5.17 所示。

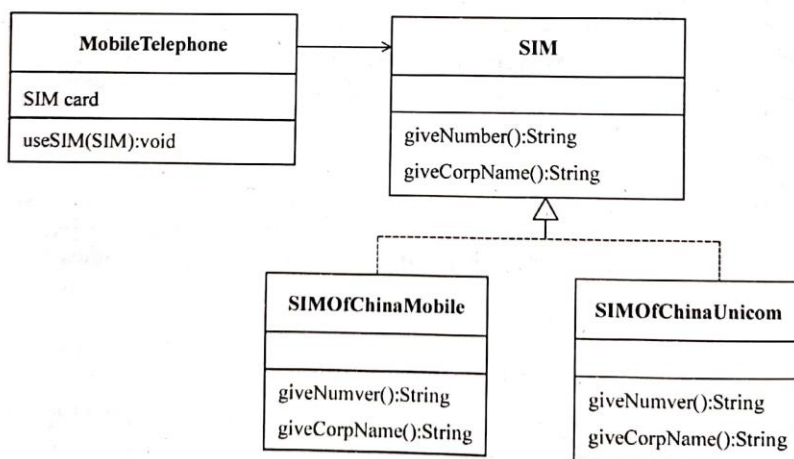


图 5.16 UML 类图

例子 13

SIM.java

```
public abstract class SIM {
```

```

public abstract void setNumber(String n);
public abstract String giveNumber();
public abstract String giveCorpName();
}

```

使用的卡是:中国移动提供的
手机号码是:13887656432
使用的卡是:中国联通提供的
手机号码是:13097656437

MobileTelephone.java

图 5.17 手机使用 SIM 卡

```

public class MobileTelephone {
    SIM card;
    public void useSIM(SIM card) {
        this.card=card;
    }
    public void showMess() {
        System.out.println("使用的卡是:"+card.giveCorpName()+"提供的");
        System.out.println("手机号码是:"+card.giveNumber());
    }
}

```

SIMOfChinaMobile.java

```

public class SIMOfChinaMobile extends SIM {
    String number;
    public void setNumber(String n) {
        number = n;
    }
    public String giveNumber() {
        return number;
    }
    public String giveCorpName() {
        return "中国移动";
    }
}

```

SIMOfChinaUnicom.java

```

public class SIMOfChinaUnicom extends SIM {
    String number;
    public void setNumber(String n) {
        number = n;
    }
    public String giveNumber() {
        return number;
    }
    public String giveCorpName() {
        return "中国联通";
    }
}

```

Application.java

```
public class Application {  
    public static void main(String args[]) {  
        MobileTelephone telephone = new MobileTelephone ();  
        SIM sim = new SIMOfChinaMobile();  
        sim.setNumber("13887656432");  
        telephone.useSIM(sim);  
        telephone.showMess();  
        sim = new SIMOfChinaUnicom();  
        sim.setNumber("13097656437");  
        telephone.useSIM(sim);  
        telephone.showMess();  
    }  
}
```

例子 13 中的类满足 5.11 节提到的“开-闭原则”，如果再增加一个 Java 源文件（对扩展开放），该源文件有一个 SIM 的子类，例如 ChinaFeiTong 子类，那么 MobileTelephone 类不需要做任何修改（对 MobileTelephone 类的修改关闭），应用程序中就可以让 telephone 对象使用 ChinaFeiTong 类提供的 SIM 卡。