Java作业3

作业1

CPU.java

```
public class CPU {
    private int speed;
    public void setSpeed(int m) {
        this.speed = m;
    }

    public int getSpeed() {
        return speed;
    }
}
```

HardDisk.java

```
public class HardDisk {
    private int amount;
    public void setAmount(int m) {
        this.amount = m;
    }
    public int getAmount() {
        return amount;
    }
}
```

PC.java

```
public class PC {
    private CPU cpu;
    private HardDisk hd;

    public void setCpu(CPU c) {
        this.cpu = c;
    }

    public void setHardDisk(HardDisk h) {
        this.hd = h;
    }

    public void show() {
```

```
        if (cpu != null && hd != null) {
            System.out.println("CPU Speed: " + cpu.getSpeed());
            System.out.println("Hard Disk Capacity: " + hd.getAmount());
        } else {
            System.out.println("CPU or Hard Disk not set.");
        }
    }
}
```

Test.java

```
public class Test {
    public static void main(String[] args) {
        // 创建一个 CPU 对象并设置速度
        CPU cpu = new CPU();
        cpu.setSpeed(2200);

        // 创建一个 HardDisk 对象并设置容量
        HardDisk disk = new HardDisk();
        disk.setAmount(200);

        // 创建一个 PC 对象
        PC pc = new PC();

        // 设置 PC 的 CPU 和 HardDisk
        pc.setCpu(cpu);
        pc.setHardDisk(disk);

        // 显示 CPU 速度和硬盘容量
        pc.show();
    }
}
```

输出结果

```
CPU Speed: 2200
Hard Disk Capacity: 200
```

作业2

Animal.java

```
public abstract class Animal {
    public abstract void cry();
```

```
    public abstract String getAnimalName();
}
```

Dog.java

```
public class Dog extends Animal {
    @Override
    public void cry() {
        System.out.println("Woof!");
    }

    @Override
    public String getAnimalName() {
        return "Dog";
    }
}
```

Cat.java

```
public class Cat extends Animal {
    @Override
    public void cry() {
        System.out.println("Meow!");
    }

    @Override
    public String getAnimalName() {
        return "Cat";
    }
}
```

Simulator.java

```
public class Simulator {
    public void playSound(Animal animal) {
        System.out.println(animal.getAnimalName() + " says: ");
        animal.cry();
    }
}
```

Application.java

```
public class Application {
    public static void main(String[] args) {
        Simulator simulator = new Simulator();
```

```
        simulator.playSound(new Dog());
        simulator.playSound(new Cat());
    }
}
```

输出结果

```
Dog says:
Woof!
Cat says:
Meow!
```

作业3

PaymentStrategy.java (抽象类)

```
public abstract class PaymentStrategy {
    public abstract double pay(double amount);
}
```

CreditCardPayment.java (具体支付方式类)

```
public class CreditCardPayment extends PaymentStrategy {
    private static final double FEE_RATE = 0.02;
    @Override
    public double pay(double amount) {
        double fee = amount * FEE_RATE;
        System.out.println("信用卡支付手续费: " + fee);
        return amount + fee;
    }
}
```

AlipayPayment.java (具体支付方式类)

```
public class AlipayPayment extends PaymentStrategy {
    private static final double DISCOUNT_RATE = 0.95;

    @Override
    public double pay(double amount) {
        double discount = amount * (1 - DISCOUNT_RATE);
        System.out.println("支付宝支付优惠: " + discount);
        return amount * DISCOUNT_RATE;
```

```
    }
}
```

WeChatPayment.java (具体支付方式类)

```
public class WeChatPayment extends PaymentStrategy {
    private static final double PROMO_CODE_DISCOUNT = 10.0; // 满减优惠

    @Override
    public double pay(double amount) {
        if (amount >= PROMO_CODE_DISCOUNT) {
            System.out.println("微信支付满减优惠: " + PROMO_CODE_DISCOUNT);
            return amount - PROMO_CODE_DISCOUNT;
        } else {
            return amount;
        }
    }
}
```

PaymentProcessor.java (处理支付请求的类)

```
public class PaymentProcessor {
    private PaymentStrategy paymentStrategy;
    // 设置当前使用的支付策略
    public void setPaymentStrategy(PaymentStrategy strategy) {
        this.paymentStrategy = strategy;
    }

    // 处理支付请求
    public double processPayment(double amount) {
        if (paymentStrategy == null) {
            throw new IllegalStateException("必须设置支付方式");
        }
        return paymentStrategy.pay(amount);
    }
}
```

Main.java (主类)

```
public class Main {
    public static void main(String[] args) {
        PaymentProcessor processor = new PaymentProcessor();

        processor.setPaymentStrategy(new CreditCardPayment());
        System.out.println("信用卡支付实际金额: " +
processor.processPayment(100));
```

```
        processor.setPaymentStrategy(new AlipayPayment());
        System.out.println("支付宝支付实际金额: " +
processor.processPayment(100));

        processor.setPaymentStrategy(new WeChatPayment());
        System.out.println("微信支付实际金额: " +
processor.processPayment(100));
    }
}
```

输出结果

```
信用卡支付手续费: 2.0
信用卡支付实际金额: 102.0
支付宝支付优惠: 5.0
支付宝支付实际金额: 95.0
微信支付满减优惠: 10.0
微信支付实际金额: 90.0
```