

Design Pattern

Decorator

```
public abstract class Beverage {
    String description = "Beverage";

    public String getDescription() {
        return description;
    }

    public abstract double cost();
}

public class Espresso extends Beverage {
    public Espresso() {
        description = "Espresso";
    }

    public double cost() {
        return 1.99;
    }
}

public class Latte extends Beverage {
    public Latte() {
        description = "Latte";
    }

    public double cost() {
        return 0.3;
    }
}

public abstract class BeverageCondimentDecorator extends Beverage {
    public abstract String getDescription();
    public abstract double cost();
}

public class Mocha extends BeverageCondimentDecorator {
    private Beverage beverage;

    public Mocha(Beverage beverage) {
        this.beverage = beverage;
    }

    public String getDescription() {
        return beverage.getDescription() + ", Mocha";
    }

    public double cost() {
        return beverage.cost() + 0.20;
    }
}
```

```

public class Soy extends BeverageCondimentDecorator {
    private Beverage beverage;

    public Soy(Beverage beverage) {
        this.beverage = beverage;
    }

    public String getDescription() {
        return beverage.getDescription() + ", Soy";
    }

    public double cost() {
        return beverage.cost() + 0.50;
    }
}

public class Whip extends BeverageCondimentDecorator {
    private Beverage beverage;

    public Whip(Beverage beverage) {
        this.beverage = beverage;
    }

    public String getDescription() {
        return beverage.getDescription() + ", Whip";
    }

    public double cost() {
        return beverage.cost() + 0.3;
    }
}

public class Application {
    public static void main(String... args) {
        Beverage beverage1 =
            new Whip(new Mocha(new Mocha(new Espresso())));
        System.out.println(beverage1.getDescription() + " - " +
            beverage1.cost());

        System.out.println();

        Beverage beverage2 = new Soy(new Latte());
        System.out.println(beverage2.getDescription() + " - " +
            beverage2.cost());
    }
}

```