



软件体系结构

《软件体系结构作业十四》

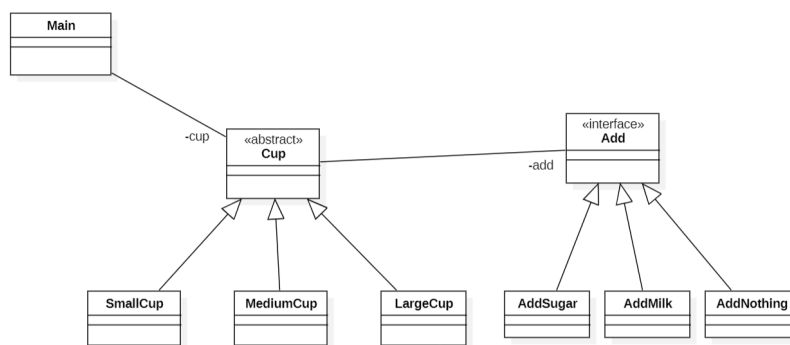
学 号 22920212204396

姓 名 黄子安

2024 年 5 月 6 日

一、试用 Bridge 模式完成下列事情：饮料的杯子有大、中、小；行为有加奶、加糖、啥都不加

根据桥接器原理将抽象部分与它的实现部分分离，使他们都可以独立的变化。这个题中杯子的大小是**实现**，可以采用**抽象类**，而加奶、加糖、不加是**行为**，可以采用**接口**，之后根据桥接器原则进行桥接从而实现两端自由组合，先给出对应的类图：



此时即可通过桥接器模式实现杯子和东西的排列组合，接下来给出代码：

Cup 抽象类代码如下，其中包含对应的 **Add** 接口作为成员，之后通过构造函数将对应的接口实现类赋值到类中，提供一个抽象方法表示饮用对应的饮料

```

public abstract class Cup {
    protected Add add;
    public Cup(Add add) {
        this.add = add;
    }
    abstract public void drink();
}
  
```

再定义对应的三个继承子类，表示大杯、中杯、小杯

```

public class MediumCup extends Cup {
    public MediumCup(Add add) {
        super(add);
    }
    public void drink() {
        System.out.println("中杯的饮料 " + add.showWhatAdded());
    }
}
  
```

```
public class LargeCup extends Cup {
    public LargeCup(Add add) {
        super(add);
    }
    public void drink() {
        System.out.println("大杯的饮料 " + add.showWhatAdded());
    }
}
```

```
public class SmallCup extends Cup {
    public SmallCup(Add add) {
        super(add);
    }
    public void drink() {
        System.out.println("小杯的饮料 " + add.showWhatAdded());
    }
}
```

再定义一个接口表示向其中加入的东西

```
public interface Add {
    String showWhatAdded();
}
```

定义具体的三个实现类，打印对应加入的东西

```
public class AddSugar implements Add {
    public String showWhatAdded() {
        return "加入糖";
    }
}
```

```
public class AddMilk implements Add {
    public String showWhatAdded() {
        return "加入牛奶";
    }
}
```

```
public class AddNothing implements Add {
    public String showWhatAdded() {
        return "不加任何东西";
    }
}
```

最后定义 Main 函数用于测试，将所有的排列组合全部打印出来

```
public class Main {
    public static void main(String[] args) {
        Cup largeCupWithMilk = new LargeCup(new AddMilk());
        Cup largeCupWithSugar = new LargeCup(new AddSugar());
        Cup largeCupWithNothing = new LargeCup(new AddNothing());
        Cup mediumCupWithMilk = new MediumCup(new AddMilk());
        Cup mediumCupWithSugar = new MediumCup(new AddSugar());
        Cup mediumCupWithNothing = new MediumCup(new AddNothing());
        Cup smallCupWithMilk = new SmallCup(new AddMilk());
        Cup smallCupWithSugar = new SmallCup(new AddSugar());
        Cup smallCupWithNothing = new SmallCup(new AddNothing());

        largeCupWithMilk.drink();
        largeCupWithSugar.drink();
        largeCupWithNothing.drink();
        mediumCupWithMilk.drink();
        mediumCupWithSugar.drink();
        mediumCupWithNothing.drink();
        smallCupWithMilk.drink();
        smallCupWithSugar.drink();
        smallCupWithNothing.drink();
    }
}
```

输出结果如下所示：

```
D:\Java17\bin\java.exe "-javaagent:D:\IdeaU\IntelliJ IDEA 2023.1.3\lib\idea_
大杯的饮料 加入牛奶
大杯的饮料 加入糖
大杯的饮料 不加任何东西
中杯的饮料 加入牛奶
中杯的饮料 加入糖
中杯的饮料 不加任何东西
小杯的饮料 加入牛奶
小杯的饮料 加入糖
小杯的饮料 不加任何东西

Process finished with exit code 0
```

可以发现原先的 9 种排列组合通过桥接器模式转换成了 6 个实现类和 2 个抽象类，当桥接器两边的子类和实现类数量较多的时候，桥接器的作用会更加显著，有效实现一个类存在两个独立变化的维度时这两个维度都可以进行独立扩展，进一步搭配组合设计模式还可以实现多个接口实现类行为共同组合，实现解耦合