

实验三实验报告 吴禹 2023214309

第一部分-策略模式

设计思路

设计一个Strategy接口，里面有个函数totalPrice，通过实现这个接口并重载函数totalPrice来实现不同的策略

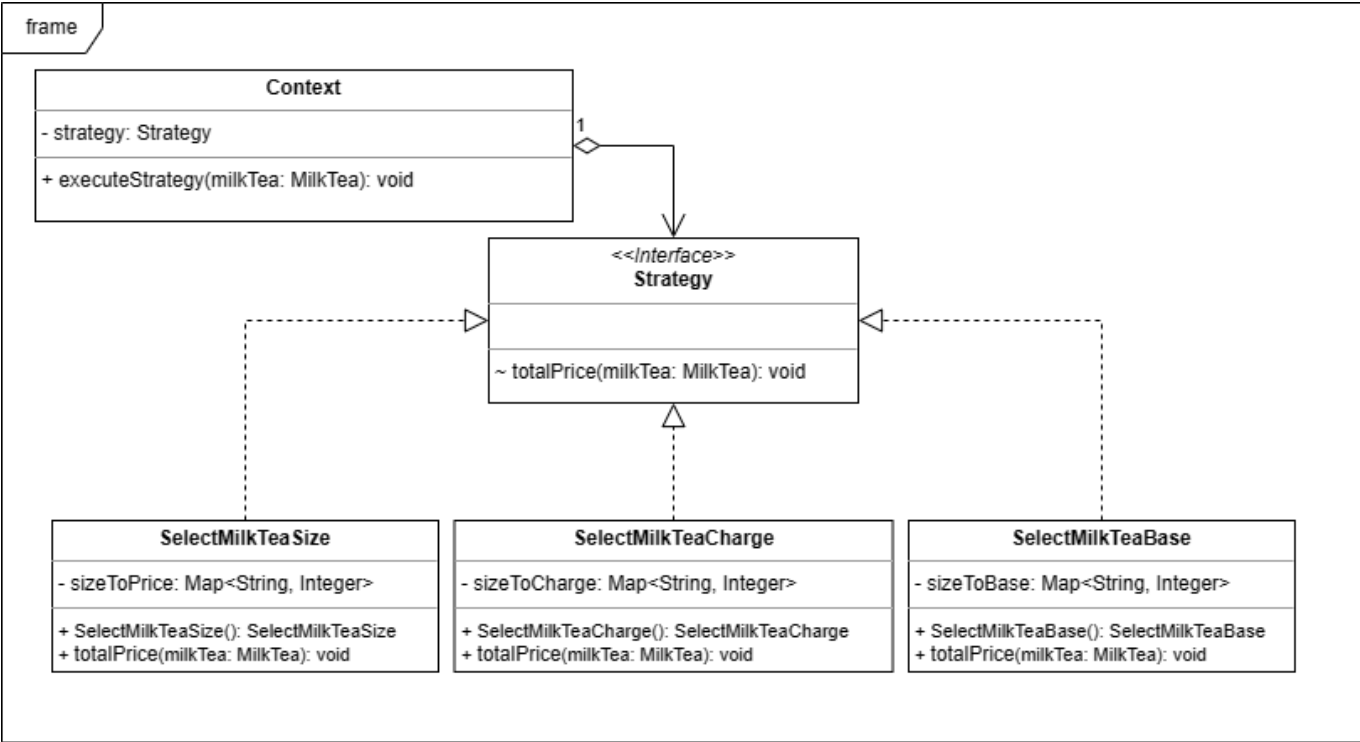
选用的设计模式

策略模式

关键类和函数的简单说明

- Strategy 抽象的策略类接口
 - totalPrice() 计算执行策略后的总价格
- SelectMilkTeaSize 选择奶茶大小的具体策略类
- SelectMilkTeaBase 选择奶茶茶底的具体策略类
- SelectMilkTeaCharge 选择奶茶加料的具体策略类
- Context 维护一个对策略对象的引用，负责将客户端请求委派给具体的策略对象执行
 - executeStrategy() 执行策略函数
- GetThePrice 用来获取价格并修改MilkTea类的类
 - getThePrice() 根据输入获得相应的价格并且据此来修改MilkTea的成员变量
- MilkTea 一个封装好的奶茶类，里面包含各种描述奶茶属性的成员变量

UML类图



测试用例

```
MilkTea milkTea = new MilkTea();
Context context = new Context(new SelectMilkTeaSize());
context.executeStrategy(milkTea);
context = new Context(new SelectMilkTeaBase());
context.executeStrategy(milkTea);
context = new Context(new SelectMilkTeaCharge());
context.executeStrategy(milkTea);
System.out.println("奶茶大小为: " + milkTea.getSize() + ", 茶底为: " + milkTea.getBase() +
    ", 加料为: " + milkTea.getCharge());
System.out.println("总价为: " + milkTea.getPrice());
```

输入输出

- 样例1

```
请选择奶茶的大小: Super: 10``Medium: 6``Large: 8``
Super
请选择奶茶的茶底: Oolong tea: 2``Black tea: 4``Green tea: 3``
Oolong tea
请选择奶茶的加料: Pearl: 1``Not add: 0``Milk cover: 3``Coconut: 2``
Pearl
请选择奶茶的加料: Pearl: 1``Not add: 0``Milk cover: 3``Coconut: 2``
Not add
奶茶大小为: Super, 茶底为: Oolong tea, 加料为: Pearl
总价为: 13
```

- 样例2

```
请选择奶茶的大小: Super: 10``Medium: 6``Large: 8``
Super
请选择奶茶的茶底: Oolong tea: 2``Black tea: 4``Green tea: 3``
long tea
错误: 请输入正确的名称。
```

第二部分-装饰模式

设计思路

设计一个抽象类MilkTeaDecorator，将MilkTea对象作为成员变量，设计decorator函数，通过继承这个抽象类MilkTeaDecorator并重载decorator函数来对MilkTea对象进行装饰

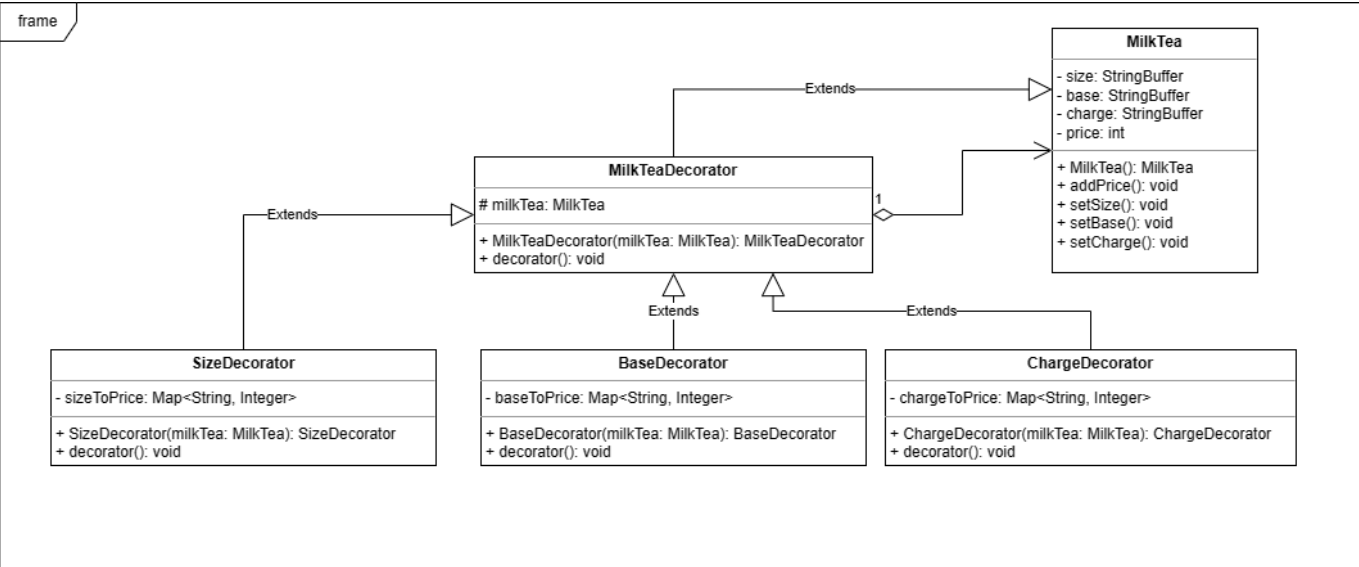
选用的设计模式

装饰模式

关键类和函数的简单说明

- MilkTeaDecorator 抽象的装饰类
 - decorator() 装饰函数
- SizeDecorator 奶茶大小装饰类
- BaseDecorator 奶茶茶底装饰类
- ChargeDecorator 奶茶加料装饰类

UML类图



测试用例

```
MilkTea milkTea = new MilkTea();
MilkTeaDecorator milkTeaDecorator = new SizeDecorator(milkTea);
milkTeaDecorator.decorator();
milkTeaDecorator = new BaseDecorator(milkTea);
milkTeaDecorator.decorator();
milkTeaDecorator = new ChargeDecorator(milkTea);
milkTeaDecorator.decorator();
System.out.println("奶茶大小为: " + milkTea.getSize() + ", 茶底为: " + milkTea.getBase() +
    ", 加料为: " + milkTea.getCharge());
System.out.println("总价为: " +milkTea.getPrice());
```

输入输出

- 样例1

```
请选择奶茶的大小: Super: 10````Medium: 6````Large: 8````
Super
请选择奶茶的茶底: Oolong tea: 2````Black tea: 4````Green tea: 3````
Black tea
请选择奶茶的加料: Pearl: 1````Not add: 0````Milk cover: 3````Coconut: 2````
Milk cover
请选择奶茶的加料: Pearl: 1````Not add: 0````Milk cover: 3````Coconut: 2````
Pearl
请选择奶茶的加料: Pearl: 1````Not add: 0````Milk cover: 3````Coconut: 2````
Not add
奶茶大小为: Super, 茶底为: Black tea, 加料为: Milk cover Pearl
总价为: 18
```

- 样例2

```
请选择奶茶的大小: Super: 10````Medium: 6````Large: 8````
Large
请选择奶茶的茶底: Oolong tea: 2````Black tea: 4````Green tea: 3````
Green tea
请选择奶茶的加料: Pearl: 1````Not add: 0````Milk cover: 3````Coconut: 2````
Not
错误: 请输入正确的名称。
```