

Decorator (Dekorátor)

Rozšiřte aplikaci „Smart Café“ o možnost vzít si nápoj s sebou nebo do vlastního hrnku s využitím vzoru Decorator. Při výpisu objednávky tak bude možné přidat k produktu [to-go] nebo [own mug] (případně obojí).

Nápověda

Pokud je potřeba, provedte úpravy třídy `CustomDrink`, aby ze třídy šlo dědit, a vytvořte abstraktní třídu `CustomDrinkDecorator`, která ze třídy dědí. Třída bude mít konstruktor, v rámci něhož bude třídě předána instance třídy `CustomDrink`, a bude přepisovat metodu `toString()`, která bude vracet `customDrink.toString()`.

Dále implementujte třídu `ToGoDecorator` a `OwnMugDecorator`, obě dědí ze třídy `CustomDrinkDecorator`. Třídy budou v rámci přepisu metody `toString()` přidávat k základnímu výpisu navíc [to-go] nebo [own mug].

Ukázka možné podoby hlavní třídy

```
EmployeeObserver waiter = new EmployeeObserver("Waiter");
EmployeeObserver barista = new EmployeeObserver("Barista");

OrderSubject order = new OrderSubject();
order.addObserver(barista);

OrderSubject payment = new OrderSubject();
payment.addObserver(waiter);

CustomDrink coffee = new CustomDrink.Builder("coffee")
    .milk()
    .sugar()
    .build();
coffee = new OwnMugDecorator(new ToGoDecorator(coffee));

CustomDrink tea = new CustomDrink.Builder("tea")
    .sugar()
    .build();
tea = new OwnMugDecorator(tea);

...
```

Pozn.: Hotová aplikace je taková, která není otestována pouze Vámi, ale především unit testy.