**TUGAS 1 DESIGN PATTERN**

**BUILDER PATTERN**

NAMA : NABILAH SHARFINA

NIM : 19104025

**Implementation**

We have considered a business case of fast-food restaurant where a typical meal could be a burger and a cold drink. Burger could be either a Veg Burger or Chicken Burger and will be packed by a wrapper. Cold drink could be either a coke or pepsi and will be packed in a bottle.

We are going to create an *Item* interface representing food items such as burgers and cold drinks and concrete classes implementing the *Item* interface and a *Packing* interface representing packaging of food items and concrete classes implementing the *Packing* interface as burger would be packed in wrapper and cold drink would be packed as bottle.

We then create a *Meal* class having *ArrayList* of *Item* and a *MealBuilder* to build different types of *Meal* objects by combining *Item*. *BuilderPatternDemo*, our demo class will use *MealBuilder* to build a *Meal*.

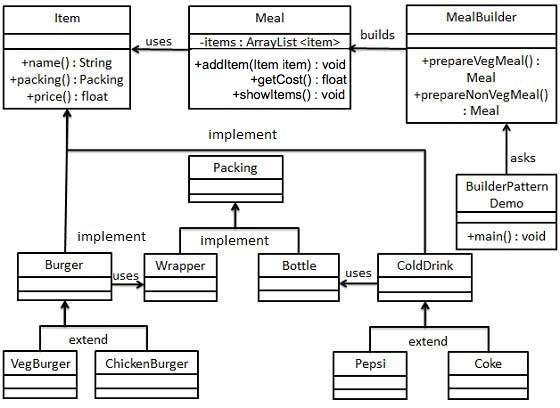
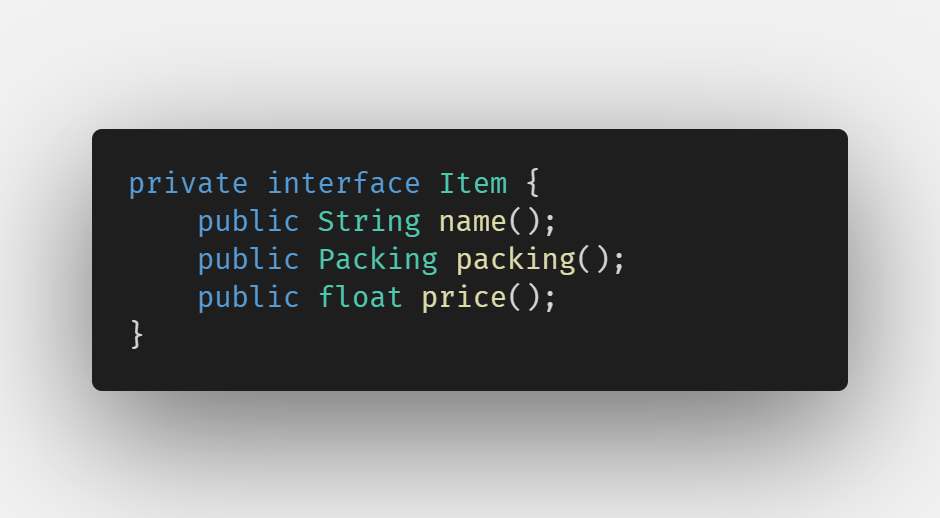


Figure 1 - Builder Pattern UML Diagram

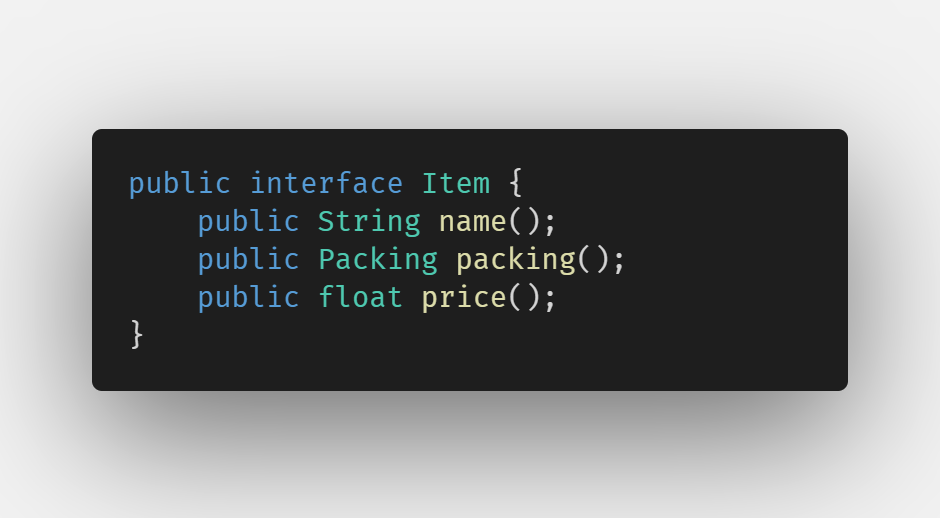
**Step 1**

Create an interface Item representing food item and packing.

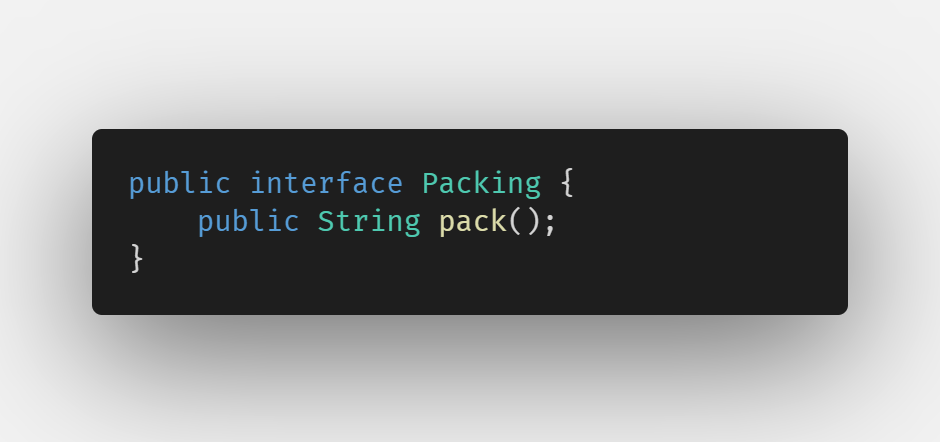
Item.java



Package private change to public:



Packing.java



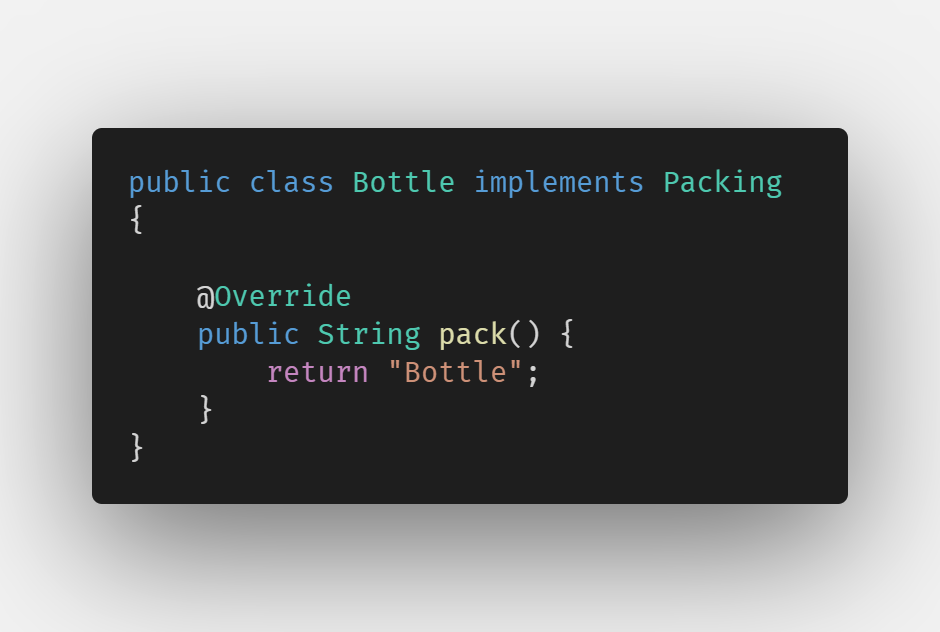
**Step 2**

Create concrete classes implementing the Packing interface.

Wrapper.java



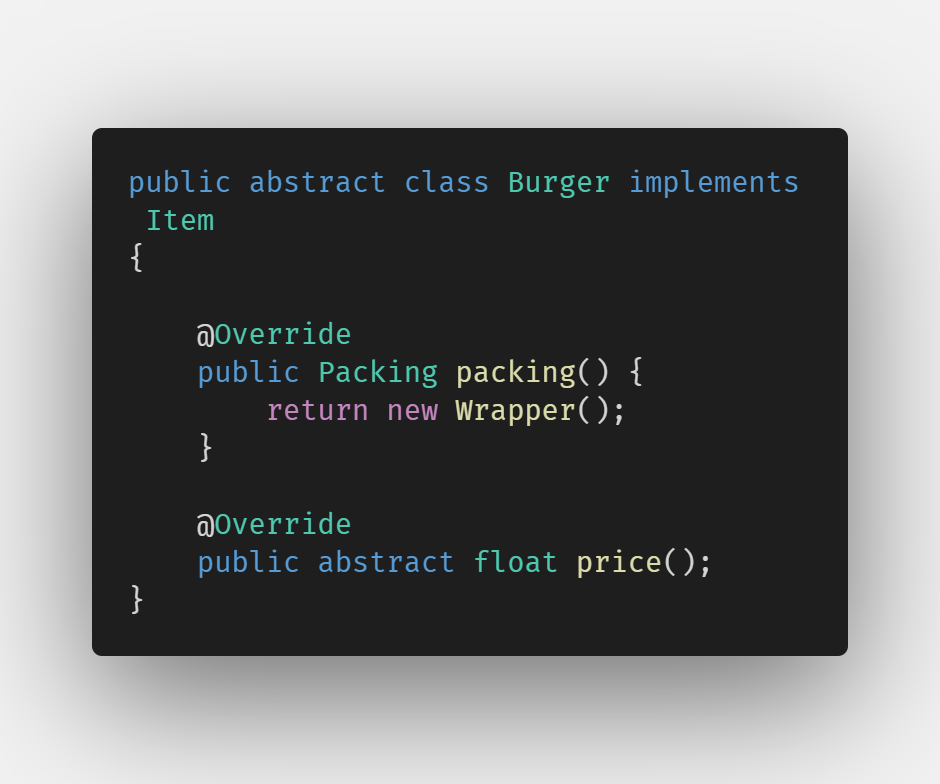
Bottle.java



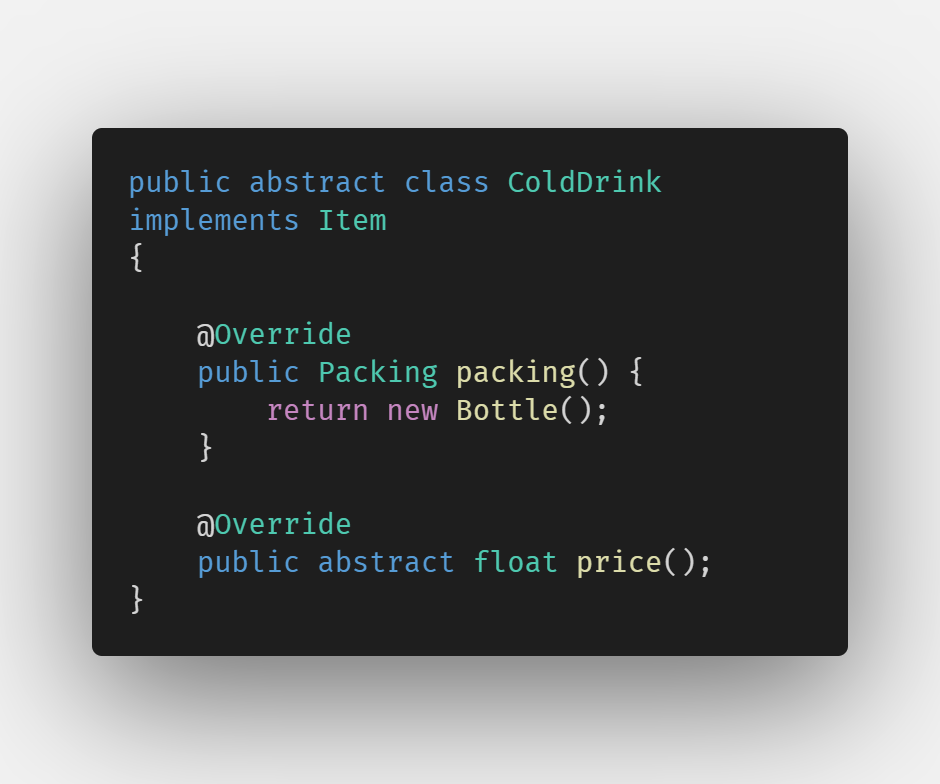
**Step 3**

Create abstract classes implementing the item interface providing default functionalities.

Burger.java



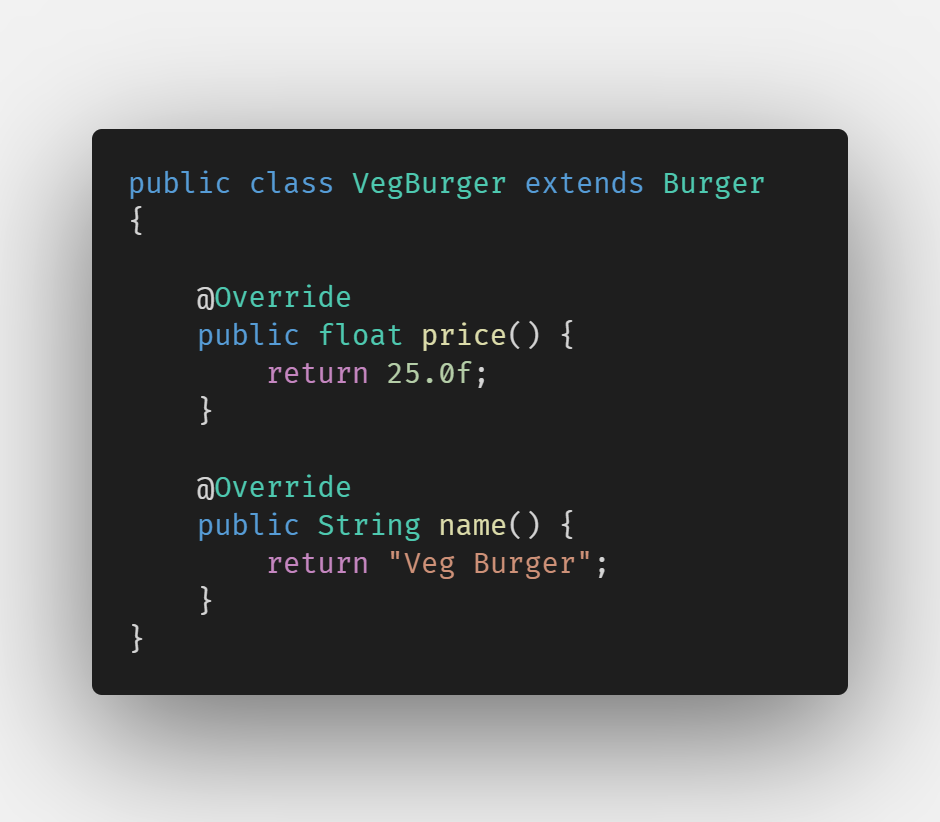
ColdDrink.java



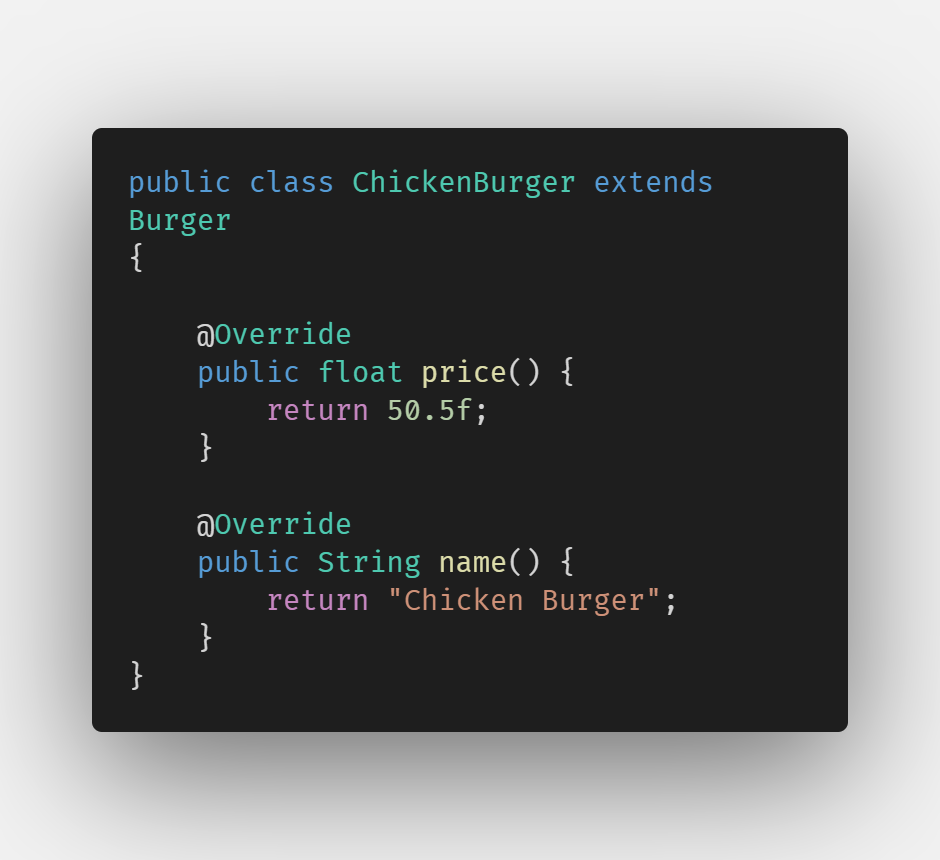
**Step 4**

Create concrete classes extending Burger and ColdDrink classes

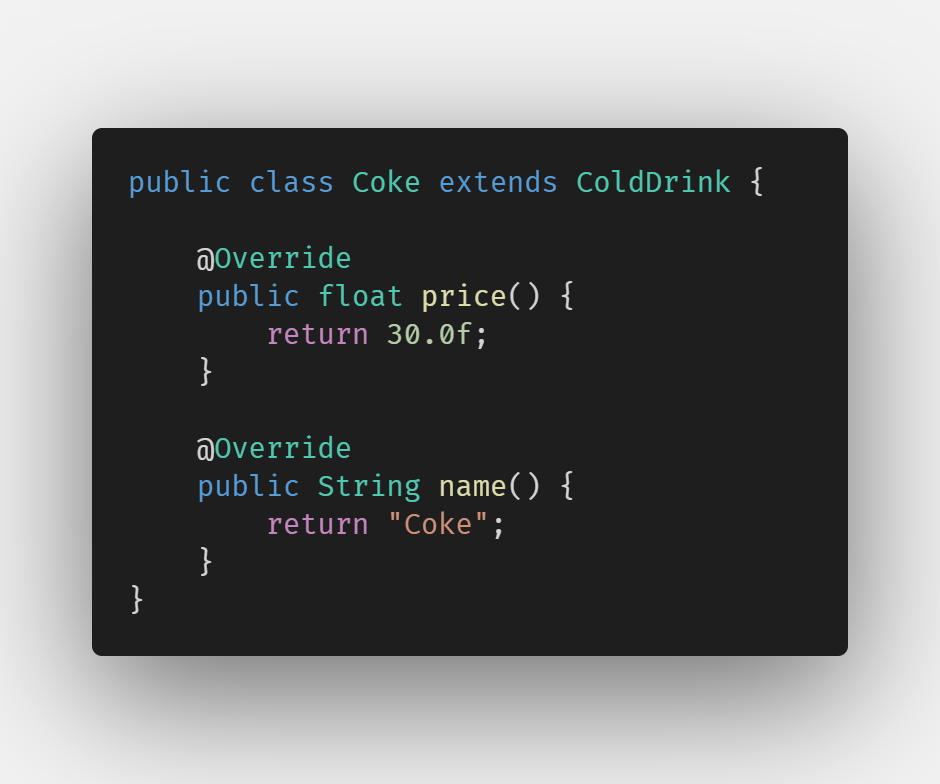
VegBurger.java



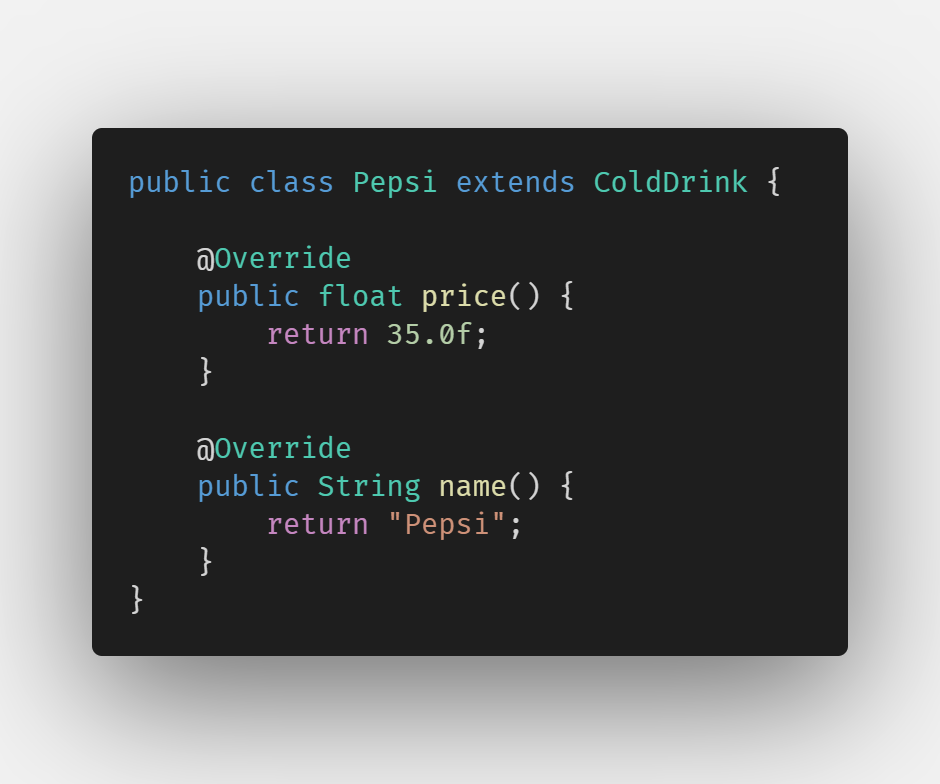
ChickenBurger.java



Coke.java



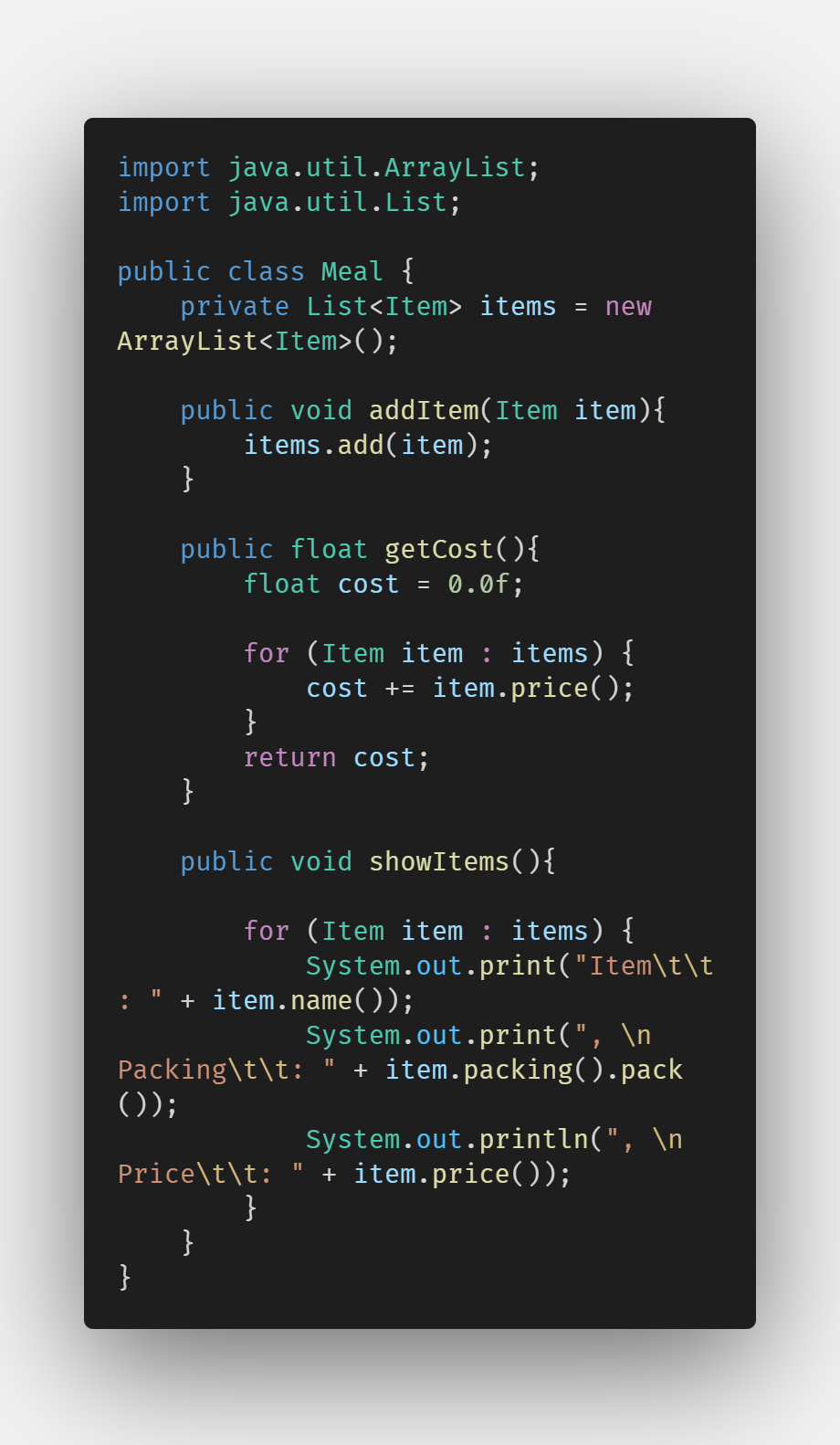
Pepsi.java



**Step 5**

Create a Meal class having Item objects defined above.

Meal.java



**Step 6**

Create a MealBuilder class, the actual builder class responsible to create Meal objects.

MealBuilder.java



**Step 7**

BuiderPatternDemo uses MealBuider to demonstrate builder pattern.

BuilderPatternDemo.java



**Step 8**

Verify the output.

