Sean Butler

CS 5800 – Adv. Software Engineering

Homework 3 – Bridge and Decorator  
github:

**Decorator**

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Food.java

1 public interface Food { 2 void Order(); 3 float Cost();

4 } 5

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Main.java

1 import java.util.Vector; 2

1. public class Main {
2. public static void main(String[] args) {
3. Customer customer = new Customer("Jason", CustomerStatus.PREMIUM);

6

1. Vector<BurgerToppings> burger\_1\_toppings = new Vector<>();
2. burger\_1\_toppings.add(BurgerToppings.CHEESE);
3. burger\_1\_toppings.add(BurgerToppings.PICKLES);
4. customer.AddBurger(burger\_1\_toppings);

11

1. Vector<BurgerToppings> burger\_2\_toppings = new Vector<>();
2. burger\_2\_toppings.add(BurgerToppings.CHEESE);
3. customer.AddBurger(burger\_2\_toppings);

15

1. customer.AddDrink(DrinkFlavour.PEPSI);
2. customer.AddFries(FriesType.WAFFLE);18

19 CustomerLoyaltyDiscount.Discount(customer);

20

1. customer.Order();
2. }
3. } 24

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Drink.java

1 public class Drink implements Food { 2 @Override

1. public void Order() {
2. System.out.println("Drink");
3. }

6

1. @Override
2. public float Cost() {
3. return 1.25f;
4. }
5. } 12

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Fries.java

1 public class Fries implements Food { 2 @Override

1. public void Order() {
2. System.out.println("Fries");
3. }

6

1. @Override
2. public float Cost() {
3. return 1.0f;
4. }
5. } 12

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Burger.java

1 public class Burger implements Food { 2 @Override

1. public void Order() {
2. System.out.println("Burger");
3. }

6

1. @Override
2. public float Cost() {
3. return 2.0f;
4. }
5. } 12

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Customer.java

1 import java.util.Vector; 2

1. public class Customer {
2. private Vector<Food> foodList;
3. private final String name;
4. CustomerStatus status;
5. private float totalCost = 0.0f;
6. private float discount = 0.0f;

9

1. Customer(String customer\_name, CustomerStatus customer\_status) {
2. name = customer\_name;
3. foodList = new Vector<>();
4. status = customer\_status;
5. }

15

1. void AddBurger(Vector<BurgerToppings> toppings) {
2. Food burger = new Burger();

18

1. for (BurgerToppings topping : toppings) {
2. if (topping == BurgerToppings.CHEESE) {
3. burger = new BurgerWithCheeseDecorator(burger);
4. }

23

1. if (topping == BurgerToppings.PICKLES) {
2. burger = new BurgerWithPicklesDecorator(burger);
3. }
4. }

28

1. foodList.add(burger);
2. }

31

1. void AddDrink(DrinkFlavour flavour) {
2. Food drink = new Drink();

34

1. if (flavour == DrinkFlavour.PEPSI) {
2. drink = new DrinkPepsiDecorator(drink);
3. }

38

1. foodList.add(drink);
2. }

41

1. void AddFries(FriesType fries\_type) {
2. Food fries = new Fries();

44

1. if (fries\_type == FriesType.WAFFLE) {
2. fries = new FriesWaffleDecorator(fries);
3. }

48

1. foodList.add(fries);
2. }

51

1. CustomerStatus GetStatus() {
2. return status;
3. }

55

1. void AddDiscount(float discount) {
2. this.discount = discount;
3. }

59

1. void Order() {
2. for (Food food : foodList) {
3. food.Order();
4. totalCost += food.Cost();
5. }

65

66 System.out.println("\nFood Total Cost for " + name + ": $" + (totalCost-discount ) + " (original cost: $" + totalCost + ")");

Page 1 of 2

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\Customer.java

67 } 68 }

69

Page 2 of 2

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\FriesType.java

1. public enum FriesType {
2. WAFFLE;
3. } 4

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\DrinkFlavour.java

1. public enum DrinkFlavour {
2. PEPSI;
3. } 4

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\FoodDecorator.java

1. public abstract class FoodDecorator implements Food {
2. protected Food foodItem;
3. protected float cost = 0.0f;

4

1. FoodDecorator(Food food\_item) {
2. foodItem = food\_item;
3. }
4. @Override
5. public void Order() { 10 foodItem.Order();
6. cost += foodItem.Cost();
7. }

13

1. @Override
2. public float Cost() {
3. return cost;
4. }
5. } 19

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\BurgerToppings.java

1. public enum BurgerToppings {
2. CHEESE,
3. PICKLES;
4. } 5

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\CustomerStatus.java

1. public enum CustomerStatus {
2. NORMAL,
3. PREMIUM,
4. ROYAL;
5. } 6

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\DrinkPepsiDecorator.java

1. public class DrinkPepsiDecorator extends FoodDecorator {
2. DrinkPepsiDecorator(Food food\_item) {
3. super(food\_item);
4. }
5. @Override
6. public void Order() { 7 foodItem.Order();

8

1. cost += foodItem.Cost();
2. cost += 0.0;

11

1. System.out.println("+ Pepsi Drink");
2. }
3. } 15

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\FriesWaffleDecorator.java

1. public class FriesWaffleDecorator extends FoodDecorator {
2. FriesWaffleDecorator(Food food\_item) {
3. super(food\_item);
4. }
5. @Override
6. public void Order() { 7 foodItem.Order();

8

1. cost += foodItem.Cost();
2. cost += 0.25;

11

1. System.out.println("+ Waffle Fries");
2. }
3. } 15

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\CustomerLoyaltyDiscount.java

1. public class CustomerLoyaltyDiscount {
2. static void Discount (Customer customer) {

3

1. if (customer.GetStatus() == CustomerStatus.PREMIUM) {
2. System.out.println("Customer Premium Discount applied: $1.0");
3. customer.AddDiscount(1.0f);
4. }
5. else if (customer.GetStatus() == CustomerStatus.ROYAL) {
6. System.out.println("Customer Royal Discount applied: $1.25");
7. customer.AddDiscount(1.25f);
8. }
9. }
10. } 14

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\BurgerWithCheeseDecorator.java

1. public class BurgerWithCheeseDecorator extends FoodDecorator {
2. BurgerWithCheeseDecorator(Food food\_item) {
3. super(food\_item);
4. }
5. @Override
6. public void Order() { 7 foodItem.Order();

8

1. cost += foodItem.Cost();
2. cost += 1.0;

11

1. System.out.println("+ Add Cheese");
2. }
3. } 15

# File - C:\msys64\home\Sean\soft\_eng\_examples\Assignment\_3\java\decorator\src\BurgerWithPicklesDecorator.java

1 public class BurgerWithPicklesDecorator extends FoodDecorator {

2

1. BurgerWithPicklesDecorator(Food food\_item) {
2. super(food\_item);
3. }
4. @Override
5. public void Order() { 8 foodItem.Order();

9

1. cost += foodItem.Cost();
2. cost += 0.5;

12

1. System.out.println("+ Add Pickles");
2. }
3. }

16