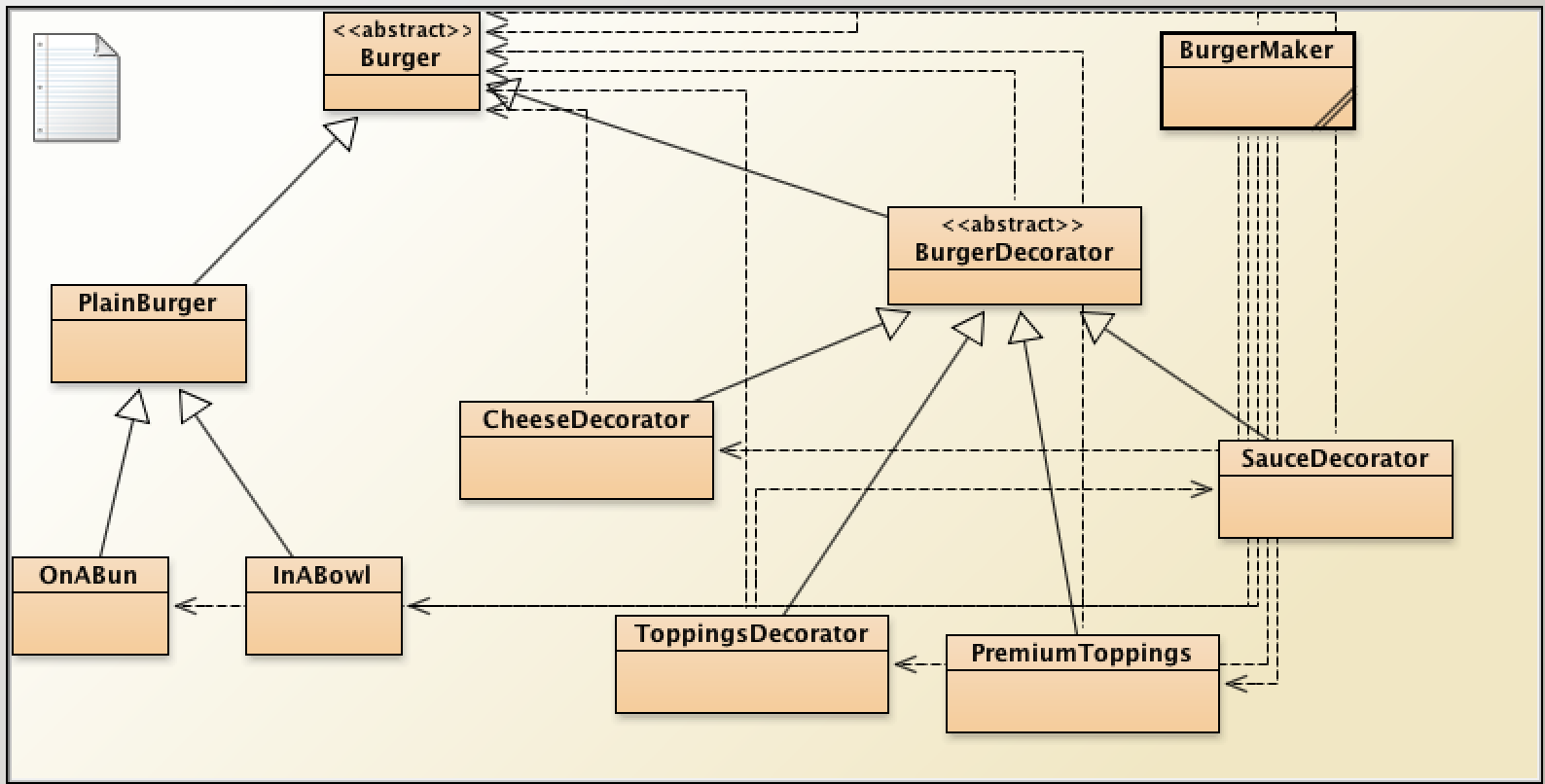
Burger Maker with Decorator Pattern



Abstract Class Burger- This class has two abstract methods. These methods are further used in “Plain Burger” class to return values such as:

getDescription()- This method returns the weight of the burger with respect to its size that may be i) small = 1/3 lbs, ii) medium = 2/3 lbs, iii) large = 1 lb

getCost() – This method returns the cost of the burger according to its base size and no other additional toppings or sauces. This method is further used in “Premium Toppings” or other such classes (as the decorator pattern) to add the cost of various toppings to the plain burger.

The code of all the classes is written below:

public abstract class Burger

{

public abstract String getDescription();

public abstract double getCost();

}

Plain Burger

public class PlainBurger extends Burger

{

String protein;

String size;

/\*\*

\* Constructor for objects of class PlainBurger

\*/

public PlainBurger(String protein, String size)

{

// initialise instance variables

this.protein = protein;

this.size = size;

}

public String getDescription()

{

// put your code here

if(size == "small")

return "1/3 lbs";

else if(size == "medium")

return "2/3 lbs";

else if(size == "large")

return "1 lb";

else

return "not a valid size";

}

public double getCost()

{

double cost = 0;

if(size == "small")

{

cost += 9.50;

}

else if(size == "medium")

{

cost += 11.50;

}

else if(size == "large")

{

cost += 15.50;

}

return cost;

}

}

On a Bun class: Child class of plain burger

public class OnABun extends PlainBurger

{

public OnABun(String protein, String size)

{

// initialise instance variables

super(protein, size);

}

public double cost()

{

System.out.println("Burger cost on a bun");

return super.getCost();

}

}

Burger Decorator: Abstract class that extends the abstract class Burger.

This class will have four child classes:

i) Cheese Decorator ii)Sauce decorator iii)Toppings Decorator iv)Premium Toppings

Each of these classes further can have many child classes such as cheddar cheese, red sauce, etc.

public abstract class BurgerDecorator extends Burger

{

public abstract String getDescription();

public abstract double getCost();

}

Cheese Decorator:

public class CheeseDecorator extends BurgerDecorator

{

// instance variables - replace the example below with your own

int i = 0;

Burger tempBurger;

public CheeseDecorator(Burger tempBurger)

{

// initialise instance variables

this.tempBurger = tempBurger;

}

public String getDescription()

{

return tempBurger.getDescription() + "adding Cheese";

}

public double getCost()

{

if (i==0)

{

i++;

return tempBurger.getCost();

}

else

{

i++;

return (tempBurger.getCost() + 1.00);

}

}

}

Sauce Decorator:

public class SauceDecorator extends BurgerDecorator

{

int i = 0;

Burger tempBurger;

public SauceDecorator(Burger tempBurger)

{

this.tempBurger = tempBurger;

}

public String getDescription()

{

return tempBurger.getDescription() + "adding Sauce";

}

public double getCost()

{

if (i==0)

{

i++;

return tempBurger.getCost();

}

else

{

i++;

return(tempBurger.getCost() + 0.75);

}

}

}

Toppings decorator:

public class ToppingsDecorator extends BurgerDecorator

{

int i = 0;

Burger tempBurger;

public ToppingsDecorator(Burger tempBurger)

{

this.tempBurger = tempBurger;

}

public String getDescription()

{

return tempBurger.getDescription() + "adding Sauce";

}

public double getCost()

{

if (i<4)

{

i++;

return tempBurger.getCost();

}

else

{

i++;

return(tempBurger.getCost() + 0.75);

}

Premium Toppings

public class PremiumToppings extends BurgerDecorator

{

// instance

Burger tempBurger;

public PremiumToppings(Burger tempBurger)

{

this.tempBurger = tempBurger;

}

public String getDescription()

{

return tempBurger.getDescription() + "adding Sauce";

}

public double getCost()

{

return(tempBurger.getCost() + 1.50);

}

}

Burger Maker Class- main method:

public class BurgerMaker

{

// instance variables - replace the example below with your own

public static void main(String args[])

{

Burger burger=new OnABun("","medium");

System.out.println(burger.getCost());

burger=new PremiumToppings(burger);

System.out.println(burger.getCost());

Burger burger1=new OnABun("","medium");

burger1= new CheeseDecorator(burger1);

burger1.getCost();

burger1= new CheeseDecorator(burger1);

System.out.println(burger1.getCost());

Burger burger2=new InABowl("","small");

burger2= new CheeseDecorator(burger2);

burger2.getCost();

burger2= new ToppingsDecorator(burger2);

burger2.getCost();

burger2= new ToppingsDecorator(burger2);

burger2.getCost();

burger2= new ToppingsDecorator(burger2);

burger2.getCost();

burger2= new ToppingsDecorator(burger2);

burger2.getCost();

burger2= new ToppingsDecorator(burger2);

System.out.println(burger2.getCost());

}

}