Notebook

# LLD: All Structural Design Patterns
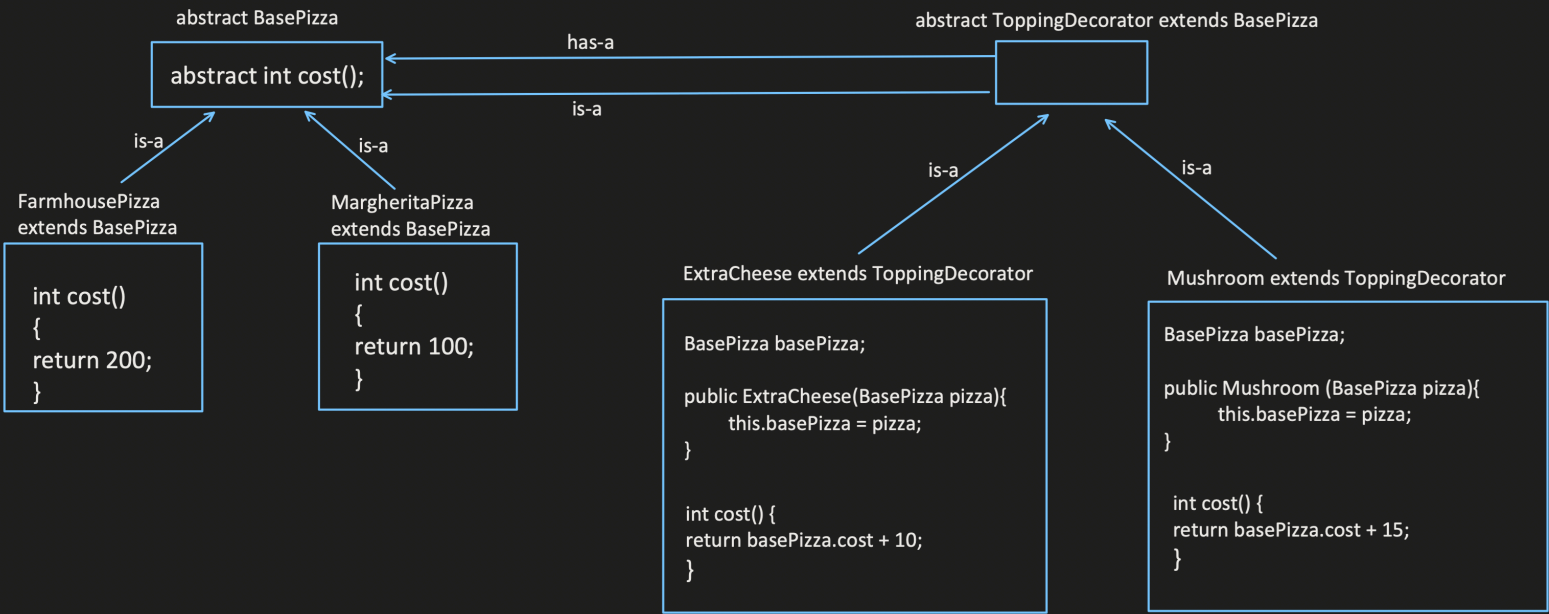
**"Concept && Coding" YT Video Notes**

*Structural Design Pattern* is a way to combine or arrange different classes and objects to form a complex or bigger structure to solve a particular requirement.

Types:
---------

1. Decorator Pattern
2. Proxy Pattern
3. Composite Pattern
4. Adapter Pattern
5. Bridge Pattern
6. Facade
7. Flyweight

## 1. Decorator Pattern:

This pattern helps to add more functionality to existing object, without changing its structure.



```
abstract BasePizza                                          abstract ToppingDecorator extends BasePizza

abstract int cost();        has-a

                            is-a

  is-a        is-a                                 is-a              is-a

FarmhousePizza      MargheritaPizza
extends BasePizza   extends BasePizza
                                              ExtraCheese extends ToppingDecorator     Mushroom extends ToppingDecorator

int cost()          int cost()
{                   {                           BasePizza basePizza;                   BasePizza basePizza;
return 200;         return 100;
}                   }                           public ExtraCheese(BasePizza pizza){   public Mushroom (BasePizza pizza){
                                                    this.basePizza = pizza;                this.basePizza = pizza;
                                                }                                      }

                                                int cost() {                           int cost() {
                                                return basePizza.cost + 10;            return basePizza.cost + 15;
                                                }                                      }
```

BasePizza pizza = new Mushroom(new ExtraCheese(new Farmhouse()));

## 2. Proxy Pattern:

This pattern helps to provide control access to original object.

```
                            interface EmployeeDao

                              void create();

                            is-a            is-a

EmployeeDaoImpl implements EmployeeDao      EmployeeDaoProxy implements EmployeeDao

  void create(){                              EmployeeDao empDaoObj;
```