Decorator Pattern

Decorator is a structural design pattern in which we try to extend the functionality of the base class object by warring it into classes of different functionalities.

For eg:

let's say we have a problem of making a system of pissa price Calculations with following constraints:

-> There is a base pizza whose price is fixed.

There are toppings which can be added on buse Pizza and these costs extra.

-> We need to take orders and calculate total, taking above into account.

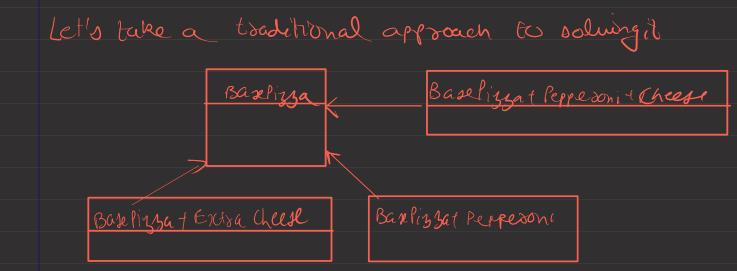
Base 11339

Toffing

Margheritta Veggie Delight Extra checse Pepperoni

so we can have



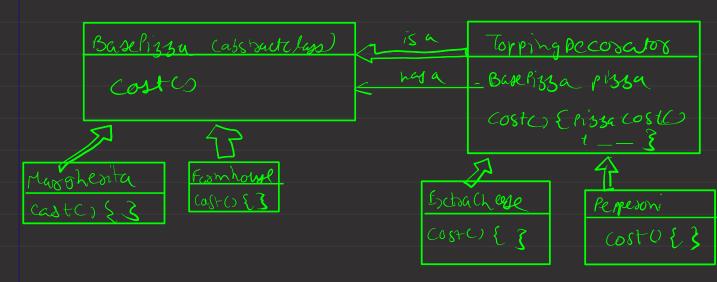


Rightnow, we only have these classes only, but imagine It yought more toppings a buses, how will the dasses grow. I how many combinations will be there

This approach leads to Class Explosion

This is where decorator pattern comes in and helps us to manage this problem by changing adding features in the run time so that we can exceed the base object functionality.

General approach with above coample



Eg code;

Baselizza p = new l'espeson (new Extracherse (
new Mosgherita()))

P.cost () >> Total cost

Applicability

- 1 Decorator pattern lets us add more functionalities to basic behaviour at suntine without breaking the cade.
- 2 voing inheritence