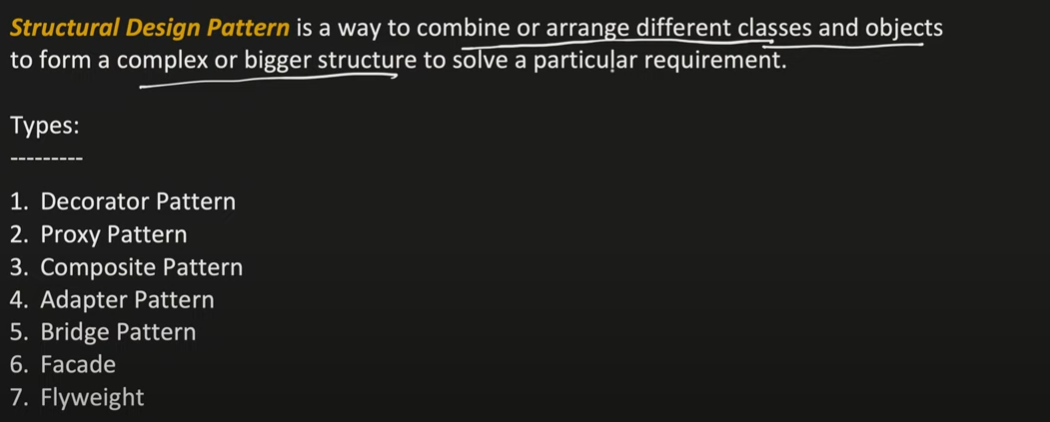
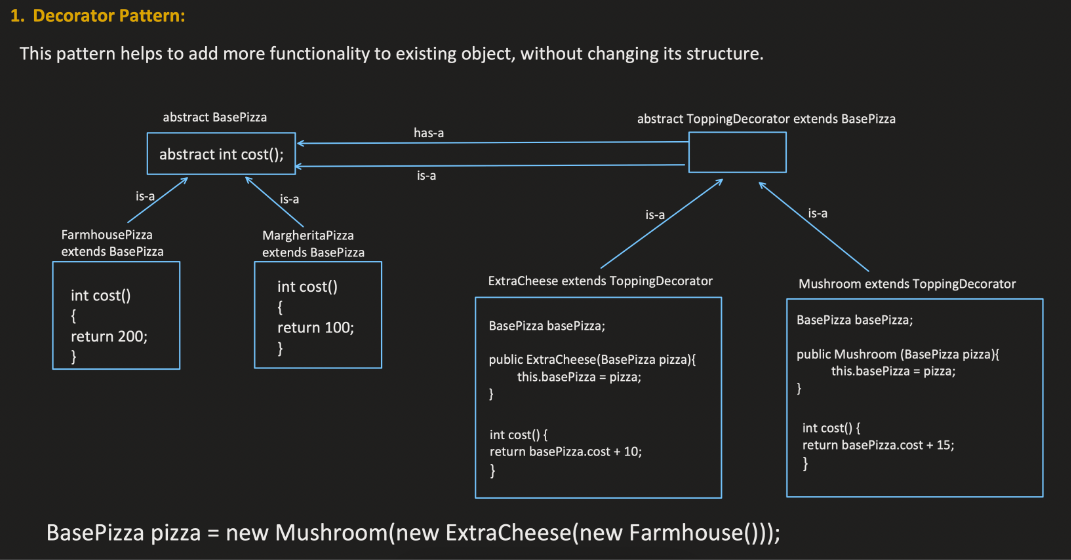
**STRUCTURAL DESIGN PATTERN**

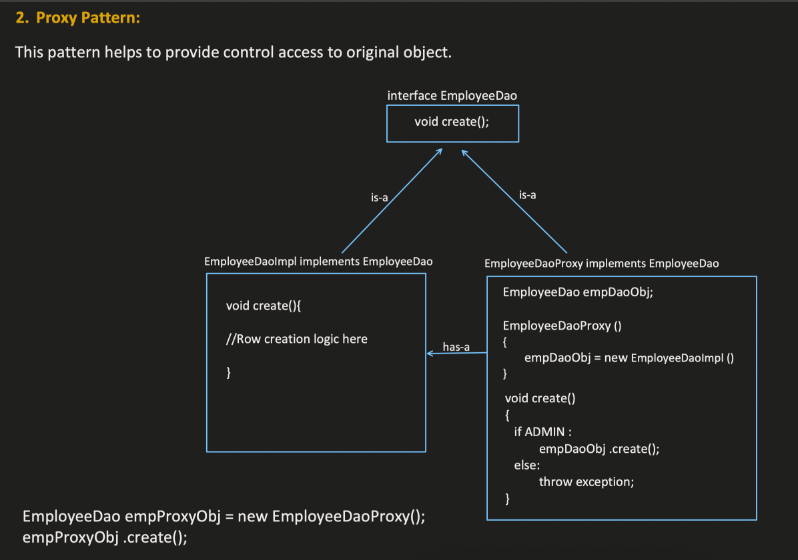


**Decorator Pattern  
  
Open to extension but close to modification**  
  
We have a has-a reln betwn ToppingDecorator and abstractPizza because inside ExtraCheese class it has a BasePizza.



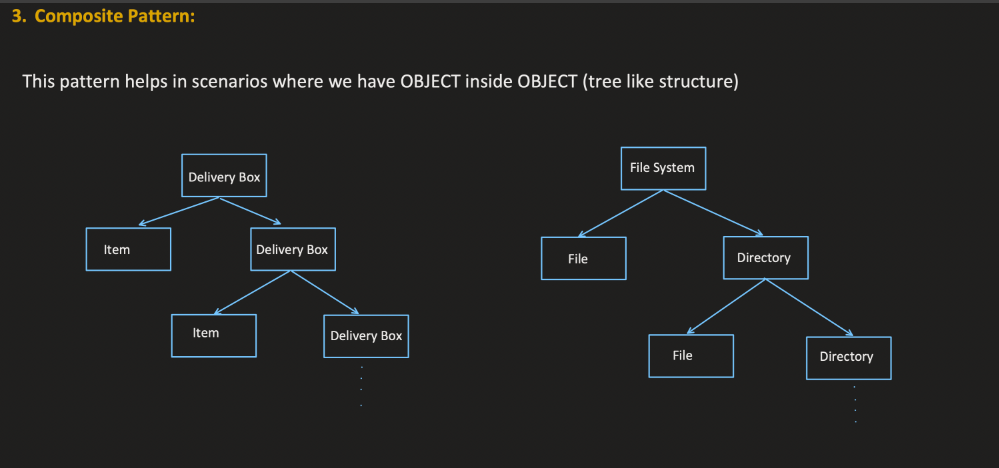
**Proxy Pattern**

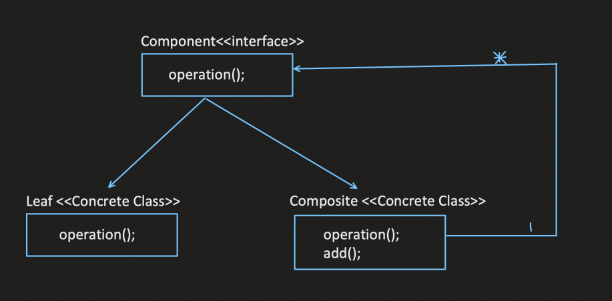
This pattern helps to provide control access to original obj. Here you can see that when we want to call the create() method of EmployeeDaoImpl , we cannot call it directly but with the help of EmployeeDaoProxy

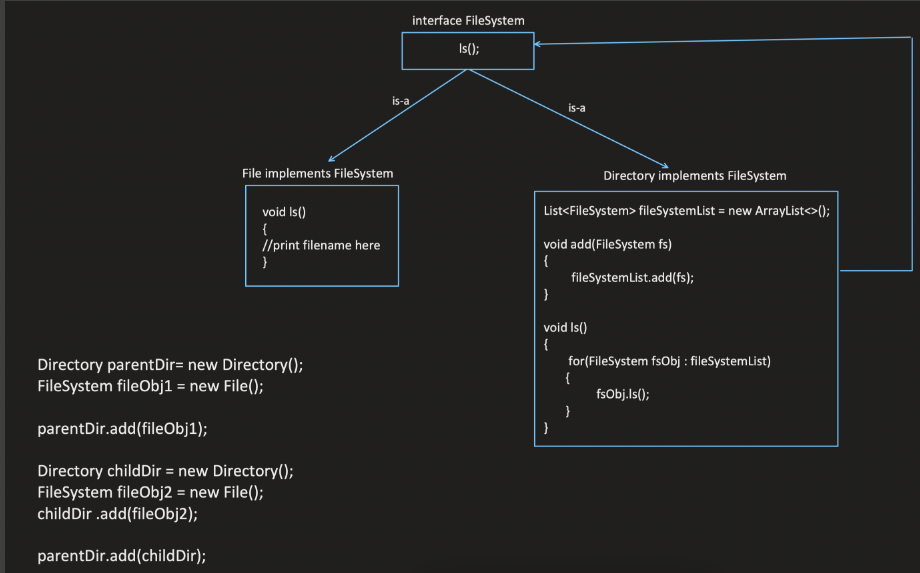


**Composite Pattern**

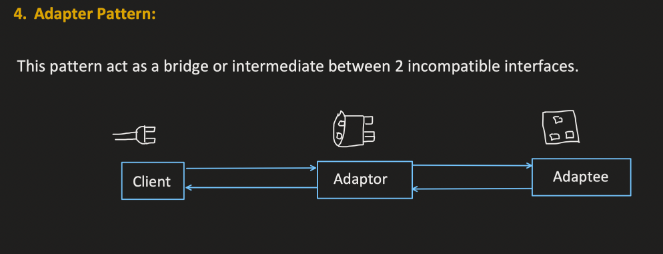
This pattern helps in scenarios when we have object with an object.

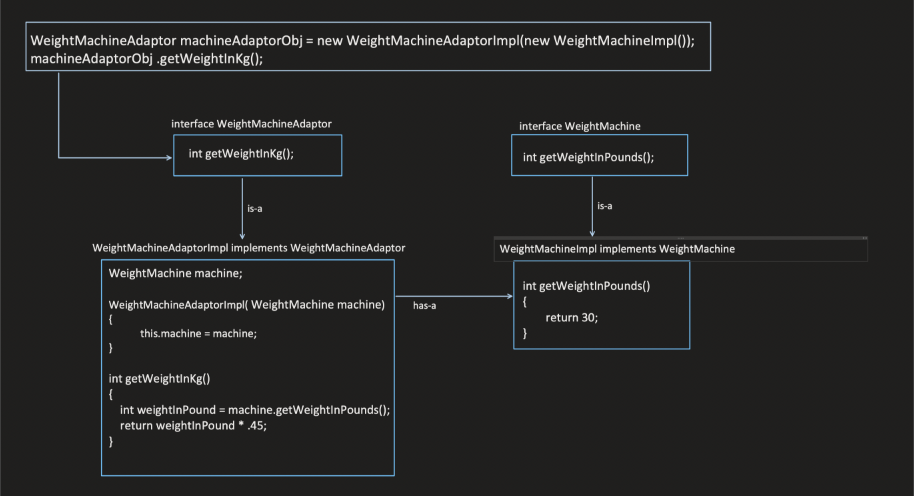




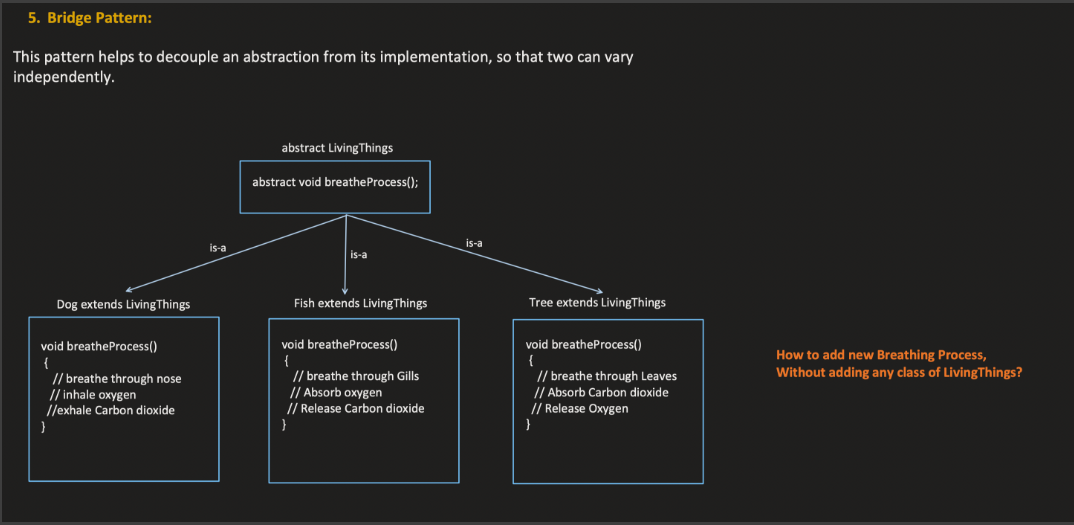


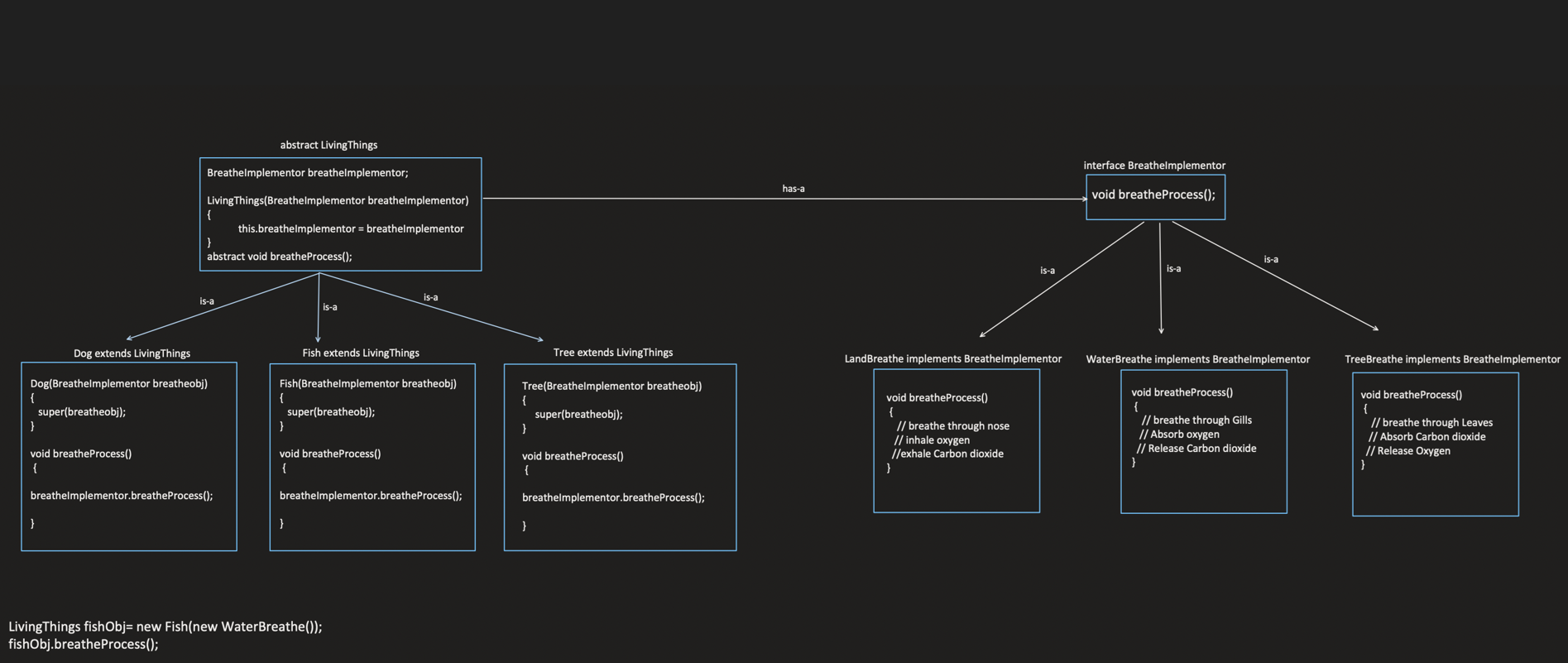
**Adapter Pattern**



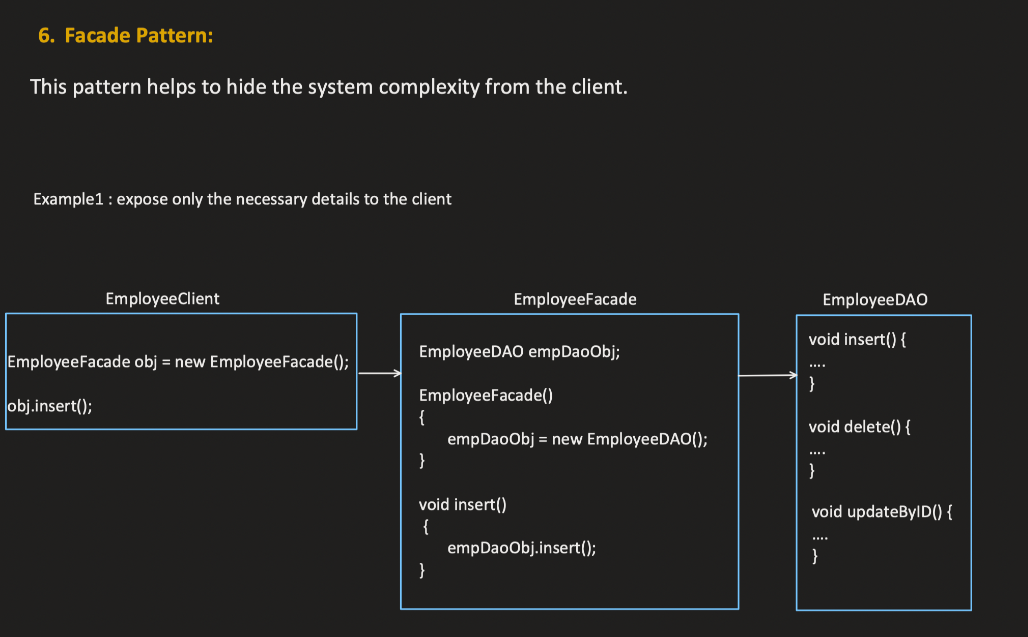


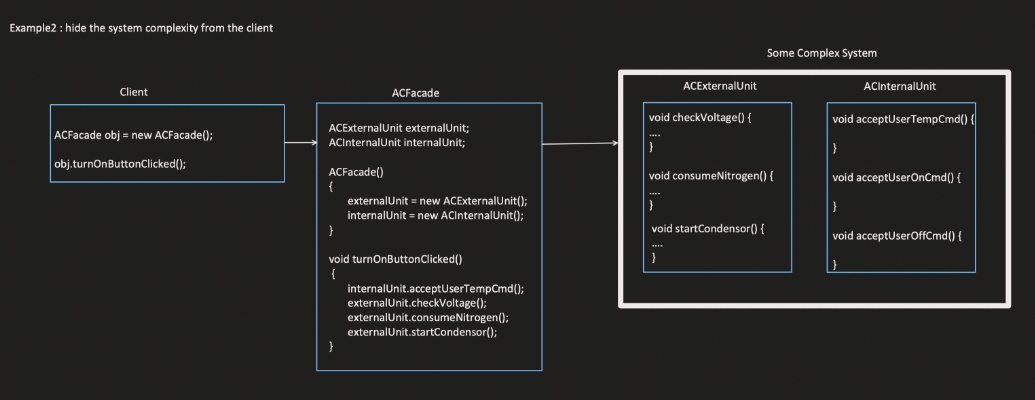
**Bridge Pattern**



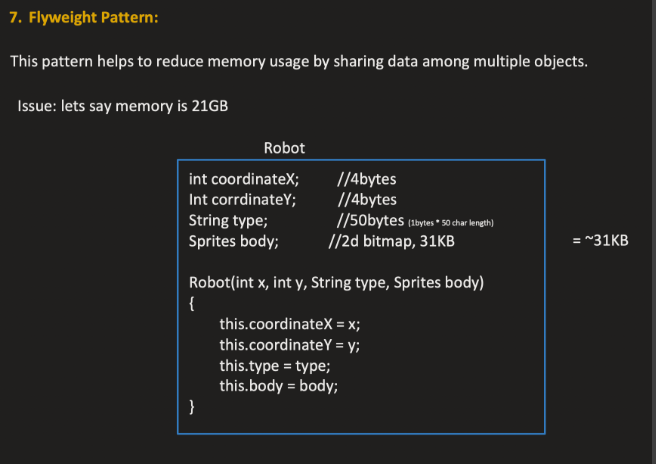


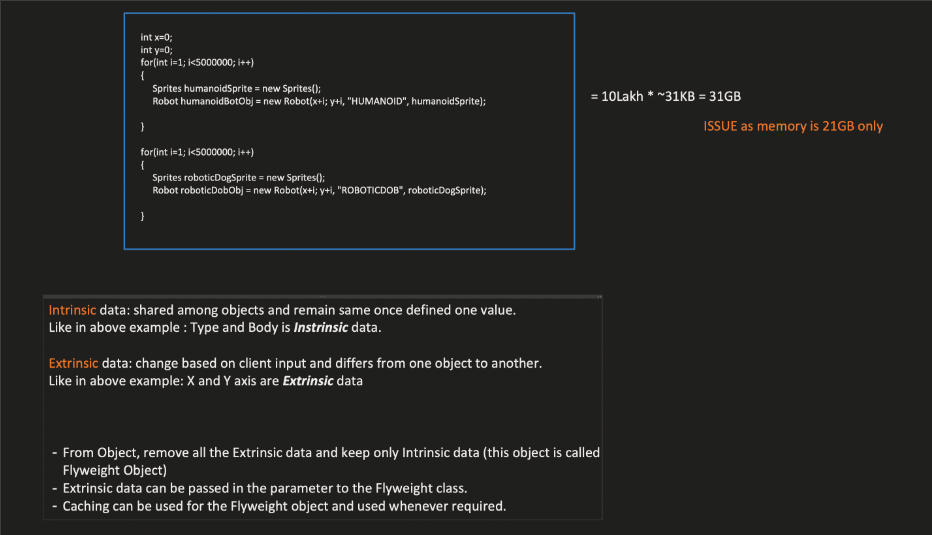
**Facade Pattern**

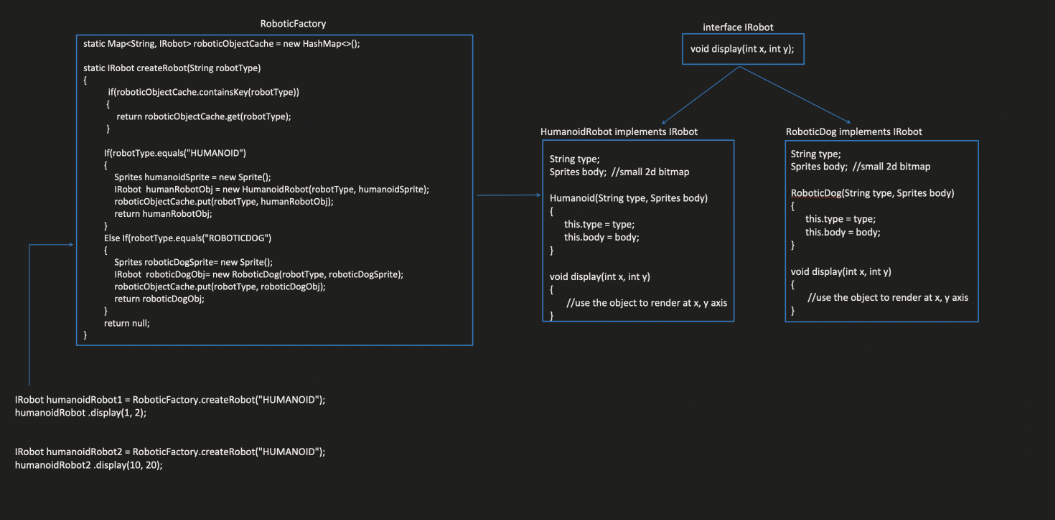




**Flyweight Pattern**

****

****

****