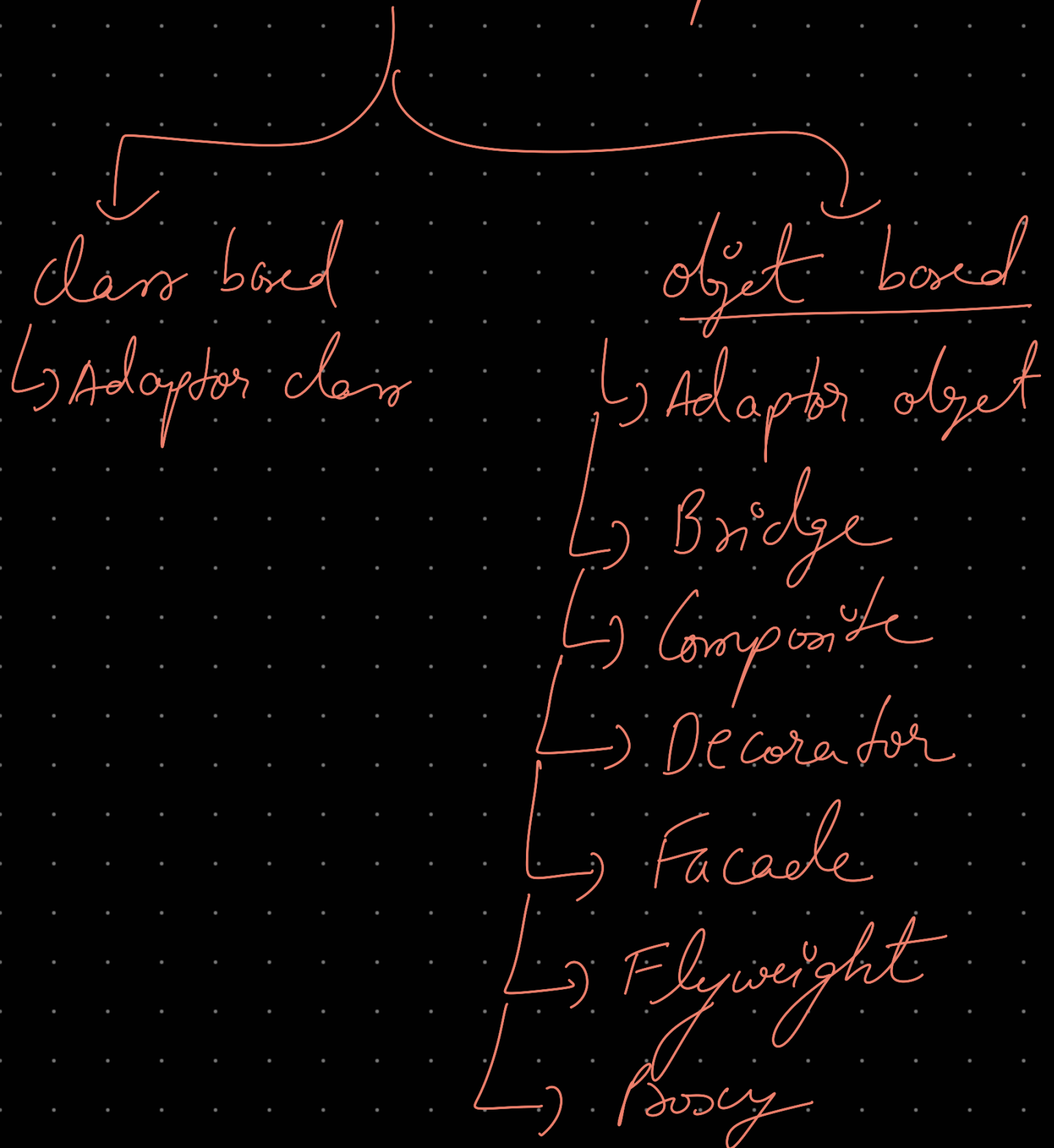


Structural Design Pattern

↳ How classes and objects are inherited or composed



[Adapter class pattern]

- ↳ Convert interface of a class into another interface
- ↳ known as Wrapper

The Adapter pattern is a structural design pattern that allows incompatible interfaces to work together. It acts as a bridge between two incompatible interfaces, enabling communication and interaction between them without modifying their source code. This pattern is particularly useful when integrating new systems with legacy systems or third-party libraries.

Key Components of the Adapter Pattern:

1. **Target Interface:** This is the interface that the client expects to work with.
2. **Adaptee Class:** This class has a method or set of methods that are incompatible with the target interface.
3. **Adapter Class:** This class implements the target interface and holds an instance of the Adaptee class. It translates calls from the target interface to the methods of the Adaptee.

