Javascript Design Patterns Lecture 2

By Mona Soliman

Agenda

- Strategy Pattern
- Observer Pattern
- What is solid principles
- Single Responsibility Principle
- Open Closed principle

Strategy Pattern:

In Strategy pattern, **a class behavior or its algorithm can be changed at run time**. This type of design pattern comes under **behavior pattern**. In Strategy pattern, we create objects which represent various strategies and a context object whose behavior varies as per its strategy object.

```
class UPS {
    calculateShipmentPackage(shipmentPackage) {
        // calculations...
        return "$45.95";
    }
}
class Fedex {
    calculateShipmentPackage(shipmentPackage) {
        // calculations...
        return "$43.20";
    }
}

class Shipping {
    setStrategy(company) {
        this.company = company;
    }
    calculate(shipmentPackage) {
        return this.company.calculateShipmentPackage(shipmentPackage);
    }
}
```

Observer Pattern:

Observer is a behavioral design pattern that **lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they're observing.**

It maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods.

```
class  YoutubeChannel{
    constructor(){
        this.videoTitle = ""
        this.subscribers = []
    }

    placeNewVideo(title){
        this.videoTitle = title
        this.notifyAll()
    }

    notifyAll(){
        return this.subscribers.forEach(sub => sub.update(this))
    }

    register(subscriber){
        this.subscribers.push(subscriber)

    }
    unregister(subscriber){
        this.subscribers =
        this.subscribers.filter(sub => sub != subscriber)
    }

}

class Person{
    update(Channl){
        // this ==> channel

        console.log(`new video uploaded ${Channl.videoTitle}`)
    }
}
```

What is SOLID principles

SOLID is an acronym for the first five object-oriented design (O
principles by Robert C. Martin (also known as Uncle Bob).

SOLID is an acronym that stands for five key design principles:

- **single responsibility principle.**
- **open-closed principle.**
- **Liskov substitution principle.**
- **interface segregation principle.**
- **dependency inversion principle.**

All five are commonly used by software engineers and provide some important benefits for developers.

Single responsibility principle

The idea behind the SRP is that every class, module, or function in a program should have one responsibility/purpose in a program. As a commonly used definition, "every class should have only one reason to change".

A class should have one, and only one, reason to change.

Open Closed principle

The idea of the principle is to design the code in a way such that new entities or functionalities can be added without modifying the existing code. A module will be said to be open if it is still available for extension. For example, it should be possible to add fields to the data structures it contains, or new elements to the set of functions it performs.

“the software entities should be open for extension but closed for modification”.

Thank you