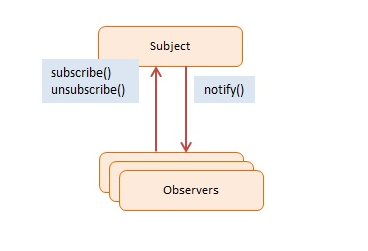
|  |
| --- |
|  |

class Observable {

constructor() {

this.observers = [];

}

subscribe(f) {

this.observers.push(f);

}

unsubscribe(f) {

this.observers = this.observers.filter(subscriber => subscriber !== f);

}

notify(data) {

this.observers.forEach(observer => observer(data));

}

}

**class** **NumberModel** {

constructor() {

this.number = 0;

this.color = 'red';

this.observers = [];

}

increment(){

const colors = ['orange','red','green','blue'];

this.number++;

this.color = colors [Math.floor(Math.random()\* colors.length)];

this.notifyObservers();

}

addObserver(o) {

this.observers.push(o);

}

unsubscribe(f) {

this.observers = this.observers.filter(subscriber => subscriber !== f);

}

notifyObservers() {

for (let o of this.observers){

o.**update**(this);

}

}

}

**class** **ElementObserver**{

constructor(elementId){

this.element = document.getElementById(elementId);

}

update(model){

this.element.innerHTML = model.number ;

this.element.style.backgroundColor = model.color ;

}

}

**class** **ConsoleObserver**{

constructor(){

}

update(model) {

console.log('The number is ' + model.number +

' and the color is ' + model.color.toUpperCase() );

}

}

**class** **HistoryObserver{**

constructor(){

this.colorHistory = [];

}

update(model){

this.colorHistory.unshift(model.color[0].toUpperCase());

let msg = 'The most recent 5 colors were: ';

for (let i = 0; i < 5 ; i++){

if (this.colorHistory[i]){

msg+= this.colorHistory[i] + ' ';

}

}

console.log(msg);

}

}

