

Modificación Patrón Observer

Objetivo

Usando el patrón Observer, tomando como base el ejemplo del administrador de configuración, implementar una nueva configuración en la clase ConfigurationManager la cual nos permita administrar configurar el formato de moneda, es decir, el usuario podrá indicar que su moneda será dólar (\$) o euro (€). Además, tendremos que crear un nuevo observador que nos permita ser notificados de que el formato ha cambiado.

Resolución

A continuación, se detallarán los pasos realizados para completar el objetivo de la tarea

1. Modificamos la clase ConfigurationManager en donde agregamos un nuevo atributo llamado currencyFormat de tipo string y los métodos getCurrencyFormat() y setCurrencyFormat(String currencyFormat)

```
public class ConfigurationManager extends AbstractObservable {

    private SimpleDateFormat defaultDateFormat;
    private NumberFormat moneyFormat;
    private String currencyFormat;

    private static ConfigurationManager configurationManager;

    private ConfigurationManager() {
    }

    public static ConfigurationManager getInstance() {
        if (configurationManager == null) {
            configurationManager = new ConfigurationManager();
        }
        return configurationManager;
    }

    public SimpleDateFormat getDefaultDateFormat() {
        return defaultDateFormat;
    }

    public void setDefaultDateFormat(SimpleDateFormat defaultDateFormat) {
        // System.out.println("Date Format change > " +
        // (this.defaultDateFormat!=null
        // ?this.defaultDateFormat.toPattern():"Null") + " to "
        // + defaultDateFormat.toPattern());
        this.defaultDateFormat = defaultDateFormat;
        notifyAllObservers("defaultDateFormat", this);
    }

    public NumberFormat getMoneyFormat() {
        return moneyFormat;
    }

    public void setMoneyFormat(NumberFormat moneyFormat) {
        // System.out.println("Date Format change > ");
        this.moneyFormat = moneyFormat;
        notifyAllObservers("moneyFormat", this);
    }

    public String getCurrencyFormat() {
        return currencyFormat;
    }

    public void setCurrencyFormat(String currencyFormat) {
        this.currencyFormat = currencyFormat;
        notifyAllObservers("currencyFormat", this);
    }
}
```

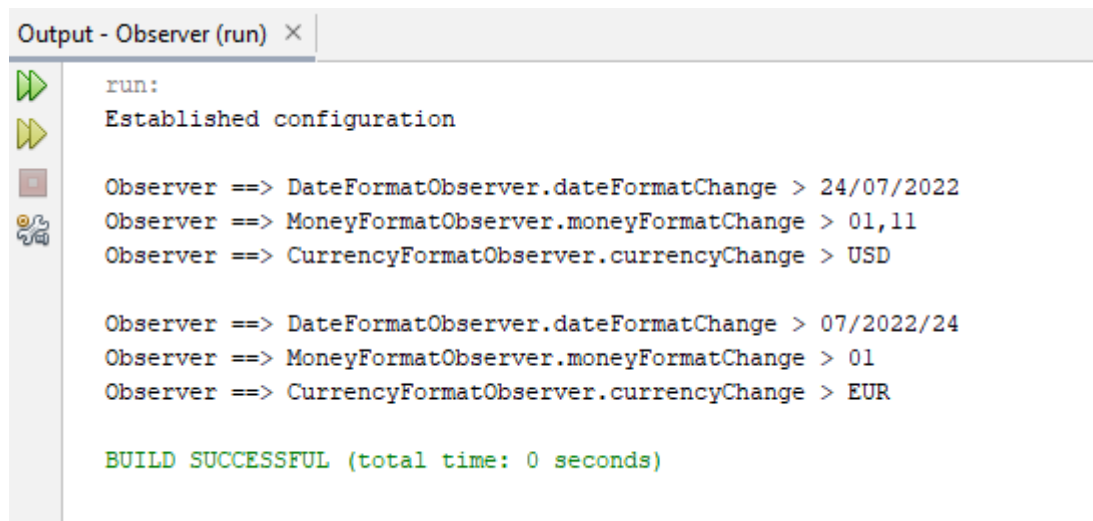
2. Creación de una clase observer para formato de moneda la cual llamaremos CurrencyFormatObserver.

```
public class CurrencyFormatObserver implements IObservable {  
    @Override  
    public void notifyObserver(String command, Object source) {  
        if (command.equals("currencyFormat")) {  
            ConfigurationManager conf = (ConfigurationManager) source;  
            System.out.println("Observer ==> CurrencyFormatObserver.currencyChange > "  
                + conf.getCurrencyFormat());  
        }  
    }  
}
```

3. Modificamos el ObserverMain en donde establecemos el cambio de currency y notificamos el mismo

```
public class ObserverMain {  
    public static void main(String[] args) {  
        ConfigurationManager conf = ConfigurationManager.getInstance();  
  
        //Se establecen los valores por default.  
        conf.setDefaultDateFormat(new SimpleDateFormat("yyyy/MM/dd"));  
        conf.setMoneyFormat(new DecimalFormat("##.00"));  
        conf.setCurrencyFormat("USD");  
        System.out.println("Established configuration");  
  
        //Se dan de alta los observers  
        DateFormatObserver dateFormatObserver = new DateFormatObserver();  
        MoneyFormatObserver moneyFormatObserver = new MoneyFormatObserver();  
        CurrencyFormatObserver currencyFormatObserver = new CurrencyFormatObserver();  
        conf.addObserver(dateFormatObserver);  
        conf.addObserver(moneyFormatObserver);  
        conf.addObserver(currencyFormatObserver);  
        System.out.println("");  
  
        //Se cambia la configuración  
        conf.setDefaultDateFormat(new SimpleDateFormat("dd/MM/yyyy"));  
        conf.setMoneyFormat(new DecimalFormat("###,##0.00"));  
        conf.setCurrencyFormat("USD");  
  
        System.out.println("");  
  
        //Se realiza otro cambio en la configuración.  
        conf.setDefaultDateFormat(new SimpleDateFormat("MM/yyyy/dd"));  
        conf.setMoneyFormat(new DecimalFormat("###,##0.00"));  
        conf.setCurrencyFormat("EUR");  
  
        conf.removeObserver(dateFormatObserver);  
        conf.removeObserver(moneyFormatObserver);  
        conf.removeObserver(currencyFormatObserver);  
        System.out.println("");  
  
        //Se realiza otro cambio en la configuración.  
        conf.setDefaultDateFormat(new SimpleDateFormat("MM/yyyy"));  
        conf.setMoneyFormat(new DecimalFormat("###,##0.00"));  
    }  
}
```

4. Después de correr el programa tenemos el siguiente resultado



```
run:
Established configuration

Observer ==> DateFormatObserver.dateFormatChange > 24/07/2022
Observer ==> MoneyFormatObserver.moneyFormatChange > 01,11
Observer ==> CurrencyFormatObserver.currencyChange > USD

Observer ==> DateFormatObserver.dateFormatChange > 07/2022/24
Observer ==> MoneyFormatObserver.moneyFormatChange > 01
Observer ==> CurrencyFormatObserver.currencyChange > EUR

BUILD SUCCESSFUL (total time: 0 seconds)
```

5. Enlace git donde se encuentra el código es <https://github.com/cristianmoreno1986/patrones>