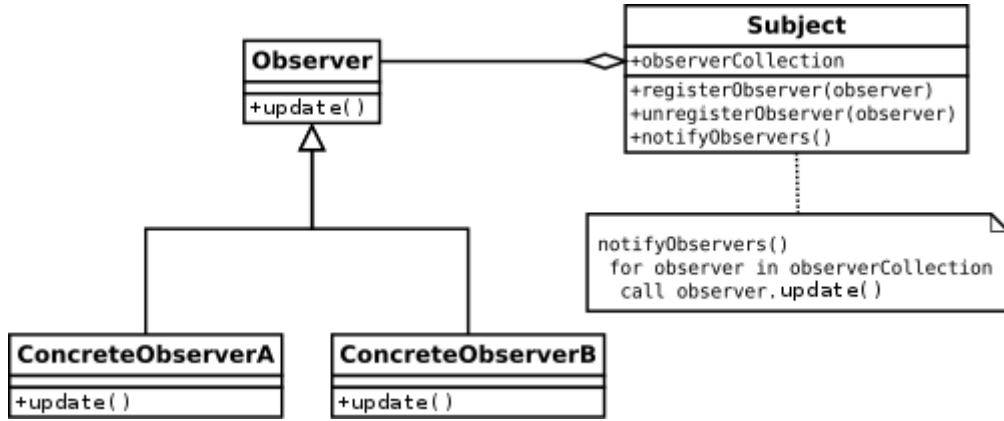


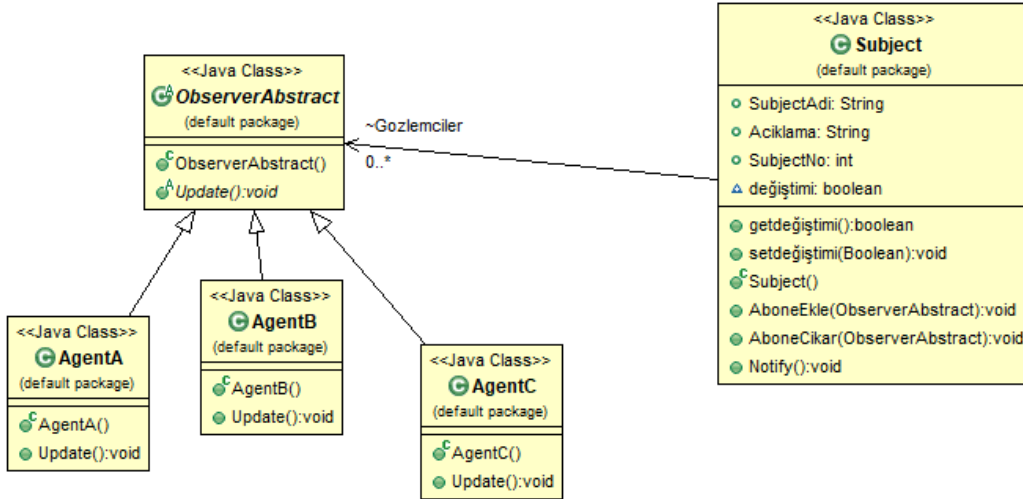
Observer Design Pattern: Genel olarak gözlemci ve gözlenen classlardan oluşan bir patterndir. Bir classdaki bağımlılıkların listesini tutmamız gerektiğinde veya değişken durumları algılamamız gerektiğinde buna başvurabiliriz.



Genel uml yapısı budur.

3 Yapı taşı vardır

- Bir abstract gözlemci class
- Ondan kalıtım alan diğer gözlemci classlar
- Gözlenen class



Observer pattern için yazdığımız kodun uml şeması bu şekilde. Görüldüğü üzere ObserverAbstracttan kalıtım alan 3 gözlemci ve bir gözlenen classtan oluşuyor.

```
1 |
2 public abstract class ObserverAbstract {
3
4
5     public abstract void Update();
6 }
7
```

Observer Abstractımız

```
1 |
2 public class AgentA extends ObserverAbstract{
3
4 @Override
5     public void Update() {
6
7         System.out.println("AgentA subjectteki değişikliği saptadı");
8
9     }
10
11 }
12
```

AgentA diğer agentlarda bununla aynıdır sadece println içeriği farklı

```

7
8 boolean deðiřtими;
9
0
1
2 public boolean getdeðiřtими() {
3
4     return deðiřtими;
5 }
6
7
8 public void setdeðiřtими(Boolean _degistimi) {
9
0     if(_degistimi ==true) {
1         Notify();
2         deðiřtими =_degistimi;
3     }
4     else {}
5         deðiřtими =_degistimi;
6
7
8 }
9
0
1 List<ObserverAbstract> Gozlemciler;
2
3 public Subject() {
4     this.Gozlemciler= new ArrayList<ObserverAbstract>();
5
6 }
7
8 public void AboneEkle(ObserverAbstract observer) {
9
0     Gozlemciler.add(observer);
1 }
2
3
4 public void AboneCikar(ObserverAbstract observer) {
5
6     Gozlemciler.remove(observer);
7 }
8
9
0 public void Notify() {
1
2
3     Gozlemciler.forEach(g->{
4
5         g.Update();
6
7     });
8
9

```

Subject kodumuzda bu řekilde

```

1
2 public class Main {
3
4     public static void main(String[] args) {
5
6         System.out.println("observer pattern örneği");
7
8         Subject subject = new Subject();
9         subject.AboneEkle(new AgentB());
10        subject.AboneEkle(new AgentA());
11        subject.AboneEkle(new AgentC());
12
13        subject.SubjectAdi = "subject1";
14
15        subject.Aciklama="subject1 subjecti";
16
17        subject.SubjectNo =65;
18
19
20        subject.setdegislimi(true);
21
22
23
24
25
26
27
28    }
29
30 }
31

```

Main kodu

```

<terminated> Main (1) [Java Application] C:\Users\fatih\.p2\pool\plugin
observer pattern örneği
AgentB subjectteki değişikliği saptadı
AgentA subjectteki değişikliği saptadı
AgentC subjectteki değişikliği saptadı

```

Çıktı

Okul bitiyor (;