

## Structural - Facade Design Pattern

### Facade design pattern'i aynı zamanda Singleton olarak da ayarlanmalıdır.

Subsystem'ler içerisinde ki class'ları wrap ederek client tarafından kullanılmasını sağlar ve kolaylaştırır. Subsystem'in içerisinde ki tüm class'lar Facade nesnesi içerisinde yer almak zorundadır. İstemciler subsystem'lere direk ulaşamıyorlar. Facade ile subsystem'ler arasında loosed coupling bir ilişki vardır. Bankacılık uygulamaları için çok uygundur. Crosscutting concern objelerinde kullanılabilir. Service Oriented içinde kullanılabilir.

```
class SubSystemOne{
    void methodOne(){
        System.out.println("Method One");
    }
}
class SubSystemTwo{
    void methodTwo(){
        System.out.println("Method Two");
    }
}
class SubSystemThree{
    void methodThree(){
        System.out.println("Method Three");
    }
}
```

SubSystem'leri wrap edecek facade class'ı

```
class Facade{
    private SubSystemOne subSystemOne;
    private SubSystemTwo subSystemTwo;
    private SubSystemThree subSystemThree;
    public Facade(){
        subSystemOne = new SubSystemOne();
        subSystemTwo = new SubSystemTwo();
        subSystemThree = new SubSystemThree();
    }
    void MethodA() {
        subSystemOne.methodOne();
        subSystemTwo.methodTwo();
    }
    void MethodB(){
        subSystemOne.methodOne();
        subSystemThree.methodThree();
    }
}
```

Main class:

```
public static void main(String[] args) {
    Facade facade = new Facade();
    facade.MethodA();
    facade.MethodB();
}
```

```
"C:\Program Files\Java\jdk1.8.0_231\bin\java.exe" ...
```

```
Method One
```

```
Method Two
```

```
Method One
```

```
Method Three
```

---