

T.C. SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNE YÖNELİMLİ ANALİZ VE TASARIM

PROJE ÇALIŞMASI

Hazırlayan

HÜSEYİN BURHAN BAŞARAN

G191210077

2. Öğretim A Grubu

burhan.basaran@ogr.sakarya.edu.tr

[Video](#)

[Kaynak Kodlar](#)

AKILLI CİHAZ TASARIMI

A. Kullanıcı Doğrulama ve Giriş Ekranı

Uygulama başlatıldıktan sonra, ilk önce veritabanı bağlantısı kontrol edilir ve veritabanı bağlantısı gerçekleşir ise kullanıcı giriş ekranı görüntülenir.

```
Veritabanına bağlandı!  
Kullanıcı Adı giriniz :
```

İlk önce kullanıcıdan kullanıcı adını girmesini ve ardından şifresini girmesi istenir.

```
Veritabanına bağlandı!  
Kullanıcı Adı giriniz : burhan  
Sifre giriniz : 123
```

Şifre veritabanında kontrol edilir.

```
Veritabanına bağlandı!  
Kullanıcı Adı giriniz : burhan  
Sifre giriniz : 12  
Kullanıcı adı veya şifre hatalı. Tekrar deneyiniz  
Kullanıcı Adı giriniz : |
```

Kullanıcı adı veya şifrenin hatalı girilmesi durumunda ekrana uyarı mesajı ile birlikte tekrar deneyiniz yazdırılır ve kullanıcı doğru kullanıcı adı ve şifre girene kadar bu işlem tekrarlanır.

```
Veritabanına bağlandı!  
Kullanıcı Adı giriniz : burhan  
Sifre giriniz : 12  
Kullanıcı adı veya şifre hatalı. Tekrar deneyiniz  
Kullanıcı Adı giriniz : burhan  
Sifre giriniz : 123  
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac     |  
| 3 - Sogutucu Kapat  |  
| 4 - Cıkıs           |  
-----  
Islem Giriniz: |
```

Kullanıcı adı ve şifrenin doğru girilmesi sonucunda cihazın ana menüsüne giriş yapılır.

B. Sıcaklığın Görüntülenmesi, Soğutucunun Açılıp Kapatılması

Cihaz başlatıldığında soğutucu kapalı bir şekilde başlatılır bu da sıcaklığın artmasına sebep olur ancak cihaz başlatıldığında sıcaklık değeri belirlenen kritik sıcaklık değerinin üzerinde ise cihaz otomatik olarak soğutucu çalıştırarak başlatılır.

a. Sıcaklığın Görüntülenmesi

```
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac      |  
| 3 - Sogutucu Kapat   |  
| 4 - Cıkıs            |  
-----  
Islem Giriniz: 1  
  
-----  
Sıcaklık : 24 Celsius  
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac      |  
| 3 - Sogutucu Kapat   |  
| 4 - Cıkıs            |  
-----  
Islem Giriniz:
```

İşlemin 1(bir) girilmesi durumunda ekrana cihazın o anki güncel sıcaklığı yazdırılır ve menü ekranı da tekrar ekrana yazdırılır.

b. Soğutucunun Açılması

```
-----  
Sıcaklık : 24 Celsius  
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac      |  
| 3 - Sogutucu Kapat   |  
| 4 - Cıkıs            |  
-----  
Islem Giriniz: 2  
  
-----  
Sogutucu acıldı.  
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac      |  
| 3 - Sogutucu Kapat   |  
| 4 - Cıkıs            |  
-----  
Islem Giriniz:
```

Soğutucu açma işlemi 2(iki) girilmesi ile gerçekleşir. Bu işlem soğutucu kapalı ise soğutucuyu çalıştırır ve ekrana soğutucu açıldı mesajını yazdırır. Soğutucunun açılması durumunda güncel sıcaklık düşmeye başlar.

Soğutucunun açık olduğu durumda tekrar soğutucu açılmaya çalışılırsa soğutucu zaten açık mesajı ekrana yazdırılır.

Burada 24 Celsius olan sıcaklık değeri soğutucu açık olması durumunda 18 Celsius'a düşmüştür.

```
Islem Giriniz: 1  
  
-----  
Sıcaklık : 18 Celsius  
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac      |  
| 3 - Sogutucu Kapat   |  
| 4 - Cıkıs            |  
-----  
Islem Giriniz: 2  
  
-----  
Sogutucu zaten acık.  
-----  
| 1 - Sıcaklık Goster |  
| 2 - Sogutucu Ac      |  
| 3 - Sogutucu Kapat   |  
| 4 - Cıkıs            |  
-----  
Islem Giriniz: |
```

c. Soğutucunun Kapatılması

Soğutucu kapatma işlemi 3(üç) girilmesi ile gerçekleşir. Bu işlem soğutucu açık ise soğutucuyu kapatır ve ekrana soğutucu kapatıldı mesajını yazdırır. Soğutucunun kapatılması durumunda güncel sıcaklık yükselmeye başlar.

```
Islem Giriniz: 1
-----
Sıcaklık : 40 Celsius
-----
| 1 - Sıcaklık Goster |
| 2 - Sogutucu Ac      |
| 3 - Sogutucu Kapat   |
| 4 - Cıkıs            |
-----
Islem Giriniz: 3
-----
Sogutucu zaten kapalı.
-----
| 1 - Sıcaklık Goster |
| 2 - Sogutucu Ac      |
| 3 - Sogutucu Kapat   |
| 4 - Cıkıs            |
-----
Islem Giriniz:
```

```
Islem Giriniz: 3
-----
Sogutucu kapatıldı.
-----
| 1 - Sıcaklık Goster |
| 2 - Sogutucu Ac      |
| 3 - Sogutucu Kapat   |
| 4 - Cıkıs            |
-----
Islem Giriniz: 1
-----
Sıcaklık : 34 Celsius
-----
| 1 - Sıcaklık Goster |
| 2 - Sogutucu Ac      |
| 3 - Sogutucu Kapat   |
| 4 - Cıkıs            |
-----
Islem Giriniz:
```

Soğutucunun kapalı olduğu durumda tekrar soğutucu kapatılma çalışılırsa soğutucu zaten kapalı mesajı ekrana yazdırılır.

Burada 34 Celsius olan sıcaklık değeri soğutucu kapalı olması durumunda 40 Celsius'a yükselmiştir.

Sıcaklığın belirlenen kritik sıcaklık üzerinde olması veya üzerine çıkması durumlarında ekrana kullanıcı bilgilendirme mesajı yazdırılır. Bu mesaj kullanıcı adı, sıcaklığın kritik sıcaklık üzerinde olduğunu, sıcaklık değeri ve bu duruma karşılık soğutucunun açılması durumudur.




Bu durumda soğutucu otomatik olarak açılır ve kullanıcı sıcaklık, kritik sıcaklığın altına düşene kadar soğutucuyu kapatamaz.

```
-----  
Sıcaklık : 74 Celsius  
-----  
| 1 - Sıcaklık Göster |  
| 2 - Soğutucu Aç      |  
| 3 - Soğutucu Kapat   |  
| 4 - Çıkış            |  
-----  
İşlem Giriniz:  
burhan -> Sıcaklık kritik sıcaklığın üzerinde : Sıcaklık : 75 Celsius : Soğutucu açıldı.  
1  
  
-----  
Sıcaklık : 70 Celsius  
-----  
| 1 - Sıcaklık Göster |  
| 2 - Soğutucu Aç      |  
| 3 - Soğutucu Kapat   |  
| 4 - Çıkış            |  
-----  
İşlem Giriniz: |
```

Burada sıcaklık 70 Celcius olarak belirlenen kritik sıcaklık değerinin üzerindedir. Bilgi mesajı ekrana yazdırılmış ve soğutucu açılıp sıcaklık 74 Celsius'dan 70 Celsius'a düşürülmüştür.

C. Veritabanı Görüntüsü ve İşlemleri

Users(kullanıcılar) tablosu üç sütundan oluşmaktadır. Bunlar; id, name ve pass şeklinde her kullanıcıya özgü id numarası (Primary key), yine kullanıcıya özgü(Unique key) isim(name) ve şifre(pass) barındırmaktadır.

Columns							+
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	
	id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	name	character varying	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	pass	character varying	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Users tablosundaki bütün kullanıcıları listeleme işlemi gerçekleştirilir.

1 SELECT * FROM users;			
Data Output Explain Messages Notifications			
	id [PK] integer	name character varying (50)	pass character varying (20)
1	1	burhan	123

Kullanıcı doğrulama işlemi için girilen kullanıcı adı ve şifre veritabanındaki giriş fonksiyonuna gönderilir ve burada kontrol edilir. Kullanıcı adı ve şifre doğru ise 1 yanlış ise 0 değerini döndüren integer bir fonksiyondur.



```
1 CREATE FUNCTION giris(_name character varying, _pass character varying) RETURNS integer
2     LANGUAGE plpgsql
3     AS $$
4 begin
5     if(select count(*) from users where name = _name and pass = _pass)>0 then
6         return 1;
7     else
8         return 0;
9     end if;
10 end
11 $$;
```

```
14 select * from giris('burhan','123');
```

Data Output		Explain	Messages	Notifications
	giris integer			
1	1			

Burada fonksiyon ve bu fonksiyonu çağırılması görüntülenmekte. Kullanıcı adı ve şifre doğru olduğu için 1 yani true değeri döndürülür.

D. Dependency Inversion

Bir sınıfın, metodun ya da özelliğin, onu kullanan diğer sınıflara karşı olan bağımlılığı en aza indirgenmelidir. Bir alt sınıfta yapılan değişiklikler üst sınıfları etkilememelidir.

```
public interface VeritabaniSurucu {  
    public void baglan();  
    public boolean giris(String name,String password);  
    public void sorguCalistir();  
    public void baglantiSonlandir();  
}
```

VeritabaniSurucu metodlarını içeren bir interface bulunmaktadır. Bu sınıf Soyutlama katmanını oluşturmaktadır.

```
public class VeritabaniIslemleri {  
  
    private VeritabaniSurucu veritabani;  
  
    public VeritabaniIslemleri(VeritabaniSurucu veritabani) {  
        this.veritabani = veritabani;  
    }  
  
    public void baglan(){  
        veritabani.baglan();  
    }  
  
    public boolean giris(String name,String password){  
        return veritabani.giris(name,password);  
    }  
}
```

VeritabaniIslemleri isimli yüksek seviye olan sınıf parametre olarak VeritabaniSurucu interface'ini almakta ve metodlarında bu interface metodlarını çağırılmaktadır.

```

public class PostgreSQLSurucu implements VeritabanıSurucu{

    private String url = "jdbc:postgresql://localhost:5432/proje";
    private Connection conn=null;

    @Override
    public void baglan(){
        try {
            conn = DriverManager.getConnection(url, user: "postgres", password: "123");
            if (conn != null) {
                String sql = "SELECT name, pass FROM users";

                System.out.println("Veritabanına bağlandı!");
            }
            else
                System.out.println("Bağlantı girişimi başarısız!");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public boolean giris(String name, String password){

        try {
            CallableStatement stmt = conn.prepareCall( sql: "{?= call giris(?,?)}");
            stmt.setString( parameterIndex: 2,name);
            stmt.setString( parameterIndex: 3,password);
            stmt.registerOutParameter( parameterIndex: 1,Types.INTEGER);
            stmt.execute();

            if(stmt.getInt( parameterIndex: 1)==1)
                return true;

            return false;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}

```

Burada VeritabanıSurucu interface'inden implement edilmiş bir düşük seviye sınıf olan PostgreSQLSurucu sınıfı bulunmaktadır.

Düşük seviyeli sınıflardaki değişimler direkt olarak yüksek seviyeli sınıfları etkilediğinden aralarında bir bağ olmamalı bunun yerine iki sınıf arasına soyut bir sınıf eklenmelidir.

Burada soyut sınıf VeritabaniSurucu interface'idir ve veritabanımız(PostgreSQL) bu türde oluşturulmuştur. Alt sınıf olan PostgreSQLSurucu sınıfında yapılan bir değişiklik ya da bu sınıfın değiştirilmesi üst sınıf olan VeritabaniIslemleri'ni etkilememektedir.

```
VeritabaniSurucu veritabaniSurucu = new PostgreSQLSurucu();  
VeritabaniIslemleri veritabaniIslemleri = new VeritabaniIslemleri(veritabaniSurucu);  
  
public AnaIslemPlatformu() { veritabaniIslemleri.baglan(); }
```

VeritabaniIslemleri'ndeki baglan metodu veritabanının değiştirilmesine rağmen aynı işlemleri gerçekleştirmektedir.

```
//VeritabaniSurucu veritabaniSurucu = new PostgreSQLSurucu();  
VeritabaniSurucu veritabaniSurucu = new MySQLSurucu();  
VeritabaniIslemleri veritabaniIslemleri = new VeritabaniIslemleri(veritabaniSurucu);  
  
public AnaIslemPlatformu() { veritabaniIslemleri.baglan(); }
```

E. Builder

Sıcaklık algılayıcı sınıfı içerisinde statik bir sıcaklık algılayıcı builder sınıfı oluşturulmuştur. Sıcaklık algılayıcının kurucu metodu private şeklinde ve parametre olarak builder alacak şekilde oluşturulmuştur. Bu da bu metodu sadece bu sınıf içerisinde ve bir builder oluşturarak erişilebilmesini sağlar.

```
public class SıcaklıkAlgılayıcı extends Publisher implements ISıcaklıkAlgılayıcı {

    private int sıcaklık;
    private int kritikSıcaklık;
    private int optimumSıcaklık;
    private boolean celsiusMu;

    private SıcaklıkAlgılayıcı(SıcaklıkAlgılayıcıBuilder builder){
        this.celsiusMu = builder.celsiusMu;
        setKritikSıcaklık(builder.kritikSıcaklık);
        setOptimumSıcaklık(builder.optimumSıcaklık);
        RandomSıcaklık();
    }

    public static class SıcaklıkAlgılayıcıBuilder{

        private int kritikSıcaklık;
        private int optimumSıcaklık;
        private boolean celsiusMu;

        public SıcaklıkAlgılayıcıBuilder(int optimumSıcaklık, int kritikSıcaklık){
            this.kritikSıcaklık = kritikSıcaklık;
            this.optimumSıcaklık = optimumSıcaklık;
            this.celsiusMu = true;
        }

        public SıcaklıkAlgılayıcıBuilder setKelvin(){
            this.celsiusMu = false;
            return this;
        }

        public SıcaklıkAlgılayıcıBuilder setCelsius(){
            this.celsiusMu = true;
            return this;
        }

        public SıcaklıkAlgılayıcı build() { return new SıcaklıkAlgılayıcı( builder: this); }

    }

}
```

Bu sınıfın kurucu metodu zorunlu olarak optimum ve kritik sıcaklık parametrelerini almaktadır ve sıcaklık birimi olarak otomatik celsius ayarlamaktadır. Set Celsius metodu sıcaklık birimini celsius, set Kelvin metodu ise sıcaklık birimini kelvin olarak ayarlamaktadır. Build metodu oluşturulan builderı yeni sıcaklık algılayıcı olarak, sıcaklık algılayıcının kurucu metodu çağırılmaktadır.

Kurucu metot builder tarafından gönderilen bu değişkenleri kendi değişkenlerine aktarır.

```
SicaklikAlgılayıcı sicaklikAlgılayıcı = new SicaklikAlgılayıcı.SicaklikAlgılayıcıBuilder( optimumSicaklık: 50, kritikSicaklık: 70)
    .setCelsius()
    .build();
```

Yeni bir sıcaklık algılayıcı oluşturulurken sıcaklık algılayıcının içerisinde bulunan builder sınıfına statik olduğu için direk erişim sağlanarak sıcaklık algılayıcı sınıfı çağırılır. Ardından diğer metotlar isteğe bağlı bir şekilde çağırılır ve en son build metodu çağırılarak bu sınıfın oluşması sağlanır.

F. Observer

Observer yani gözlemci diye nitelendirilen, anlamı gibi işlev gören bir tasarım desendir. Elimizdeki mevcut nesnenin durumunda herhangi bir değişiklik olduğunda, bu değişikliklerden diğer nesneleri haberdar eder.

Bu sistemde sıcaklık değerinin kritik sıcaklıktan yüksek olması durumunda kullanıcılara mesaj gönderilecek.

```
package observer;

public interface IObserver {
    public void update(String sıcaklık);
}
|
```

Gözlemci(observer) Interface'i tanımlandı. Duyurular bu inteface üzerinden sağlanmaktadır.

```
package observer;

public interface ISubject {
    public void attach(IObserver o);
    public void detach(IObserver o);
    public void notify(String m);
}
```

ISubject interface'i ise duyuru sisteminde ki kullanıcıların temsili olarak düşünebiliriz. Attach ile kullanıcıları bu sisteme bağlar, detach ile çıkarır ve notify ile duyurular gönderilmektedir.

```

package observer;

import ...

public class Publisher implements ISubject{
    private List<IObserver> subscribers = new ArrayList<>();

    @Override
    public void attach(IObserver subscriber) { subscribers.add(subscriber); }

    @Override
    public void detach(IObserver subscriber) { subscribers.remove(subscriber); }

    @Override
    public void notify(String mesaj) {
        for(IObserver subscriber: subscribers) {
            subscriber.update(mesaj);
        }
    }
}

```

Publisher(yayımcı) sınıfı ile ISubject interface'ini genişlettik ve IObserver türünde bir liste oluşturduk. Bu listeye ekleme ve çıkarma işlemlerinden yanı sıra bilgilendirme metodu ile bütün kullanıcılara gönderilen mesajın iletilmesi sağlanır.

```

public class Kullanici implements IObserver{

    private String name;

    public Kullanici(String name) { this.name = name; }

    @Override
    public void update(String mesaj) { System.out.print("\n" + name + " -> " + mesaj ); }
}

```

Kullanıcı sınıfı IObserver interface'ini implement etmiş, yani artık birer gözlemci ve duyuru yapıldığında update() metoduna gelen parametre ile birlikte kullanıcı adını da duyurmaktadır..

```

public boolean kritikSicaklikKontrol(){
    if(sicaklik>=kritikSicaklik){
        notify( mesaj: "Sıcaklık kritik sıcaklığın üzerinde : " + sicaklikOku() + " : ");
        return true;
    }
    else
        return false;
}

```

Sıcaklık algılayıcı sınıfı Publisher'ı extends etmiş ve sıcaklığın kritik sıcaklıktan yüksek olması durumunda notify metodu ile kullanıcılara uyarı mesajı ile birlikte okunan sıcaklık değerini göndermektedir.

```

Kullanici s1 = new Kullanici(name);
sicaklikAlgilyici.attach(s1);

```

Ana işlem platformu sınıfı ile yeni bir kullanıcı, kullanıcı adı ile birlikte eklenir ve bu eklenen kullanıcı attach metodu ile bağlanır.

```

@Override
public boolean sogutucuKapat() {
    if(sicaklikAlgilyici.kritikSicaklikKontrol()){
        sicaklikAlgilyici.sicaklikAzalt();
        return eyleyici.sogutucuAc();
    }
    sicaklikAlgilyici.sicaklikArttir();
    return eyleyici.sogutucuKapat();
}

```

Soğutucu kapat metdodu sıcaklık algılayıcının kritik sıcaklık kontrol metodunu çağırması durumunda uygun koşullar sağlandığında kullanıcı bilgilendirme işlemi gerçekleşir.

```

burhan -> Sıcaklık kritik sıcaklığın üzerinde : Sıcaklık : 75 Celsius : Sogutucu acıldı.

```


VIDEO:

<https://youtu.be/nsZ4SKNuHSA>

GİTHUB:

<https://github.com/burhan154/cs-sakarya/tree/master/BSM%20204%20-%20Nesne%20Y%C3%B6nelimli%20Analiz%20ve%20Tasar%C4%B1m>