

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

NESNE YÖNELİMLİ ANALİZ VE TASARIM DERSİ

PROJE ÇALIŞMASI

NESNELERİN İNTERNETİ SİSTEMLERİ İÇİN AKILLI CİHAZ TASARIMI

Hazırlayan

ADEM YILMAZ

G191210305

2.ÖĞRETİM B GRUBU

adem.yilmaz10@ogr.sakarya.edu.tr

Kaynak Kod Link: <https://github.com/ademyilmz34/Projem>

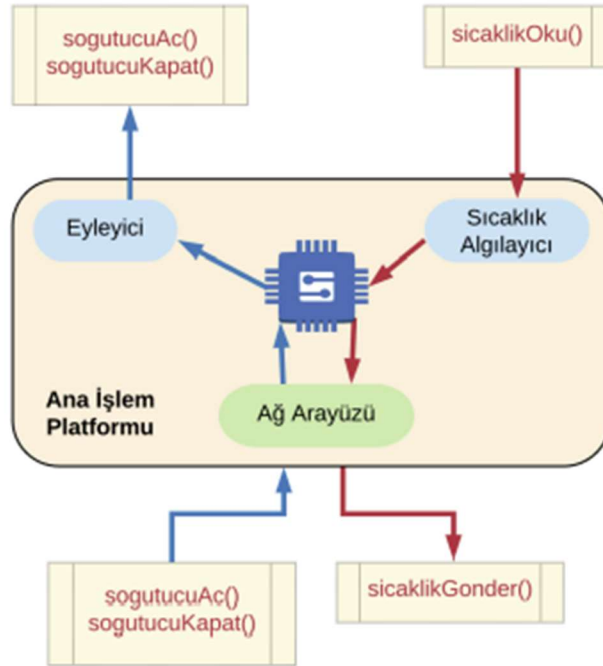
PROJE TANITIMI

Açıklama

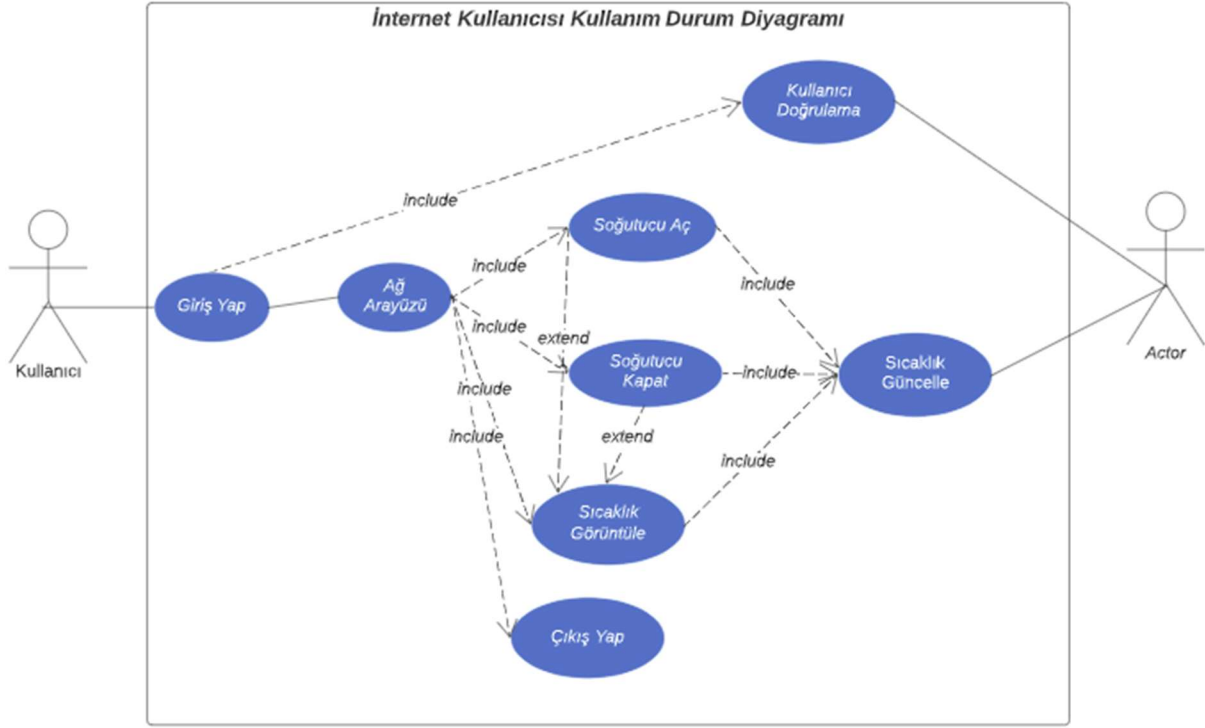
Elimizde, elektromekanik sistemi kullanılabilir durumda olan, fakat denetleyicisi çalışmayan bir soğutucu bulunmaktadır. Bu soğutucuyu internet üzerinden kontrol etmek üzere, aşağıda ana hatları verilen bir akıllı cihaz üretilmiştir. Çalışma kapsamında, bu cihaz için geliştirilmesi beklenen yazılımın; analizini, tasarımını ve gerçeklemesini yapmanız beklenmektedir.

Akıllı Cihazın Tanıtımı

- **Akıllı cihaz**, şekilde görüldüğü gibi, ana işlem platformu ile çevresel birimlerden (eyleyici, sıcaklık algılayıcısı ve ağara yüzü) oluşmaktadır.
- **Ana işlem platformu** geliştirilecek yazılımın çalıştırılacağı birimdir. Çevre birimleri ile şekilde verilen ara yüzler üzerinden haberleşmektedir.
- **Sıcaklık algılayıcı** modül ortam sıcaklığını ölçer.
- **Eyleyici modül** soğutucunun açılması ve kapatılması işlemlerini yerine getirir.
- İnternet kullanıcıları **ağ arayüzünü** kullanarak; sıcaklık görüntüleme, soğutucuyu açma ve soğutucuyu kapatma işlemlerini yerine getirebilirler.
- **Akıllı cihaz**; kapalı, açılış testi yapıyor, bekleme, algılama, servis dışı ve işlem yapıyor gibi durumlara sahiptir.



İnternet Kullanıcısı için Kullanım(Use-Case) Diyagramı



Kullanım Durumu: Sıcaklığın Görüntülenmesi

Hazırlayan: Adem Yılmaz

Sürüm: v1.0

Tarih: 19/04/2022

İlgili Aktörler: İnternet Kullanıcısı, Soğutucu Bilgi Sistemi

Giriş Koşulu:

- Kullanıcı sisteme giriş yapmış olmalıdır.
- Cihaz açık ve kullanılabilir durumda olmalıdır.

Çıkış Koşulu: Kullanıcı işlemi tamamlanmıştır.

Ana Olay Akışı(Ana Senaryo Başarılı)

1. Soğutucu sistemi, ekrana kullanıcıdan kullanıcı adını girmesini isteyen bir mesaj yazdırır.
2. Kullanıcı, tuş takımını kullanarak kullanıcı adını girer.
3. Kullanıcı adı doğrulaması için Soğutucu Bilgi Sistemine istek gönderilir.
4. Sistem isteği kabul eder ve ekrana kullanıcıdan şifre girilmesini isteyen mesaj gönderilir.
5. Kullanıcı, tuş takımını kullanarak şifreyi girer.
6. Kullanıcı doğrulaması için soğutucu sistemine istek gönderilir.
7. Soğutucu sistemi erişim isteğini kabul eder ve arayüze erişim sağlanır.
8. Kullanıcı, arayüzden tuş takımı yardımıyla sıcaklık görüntüleme seçeneğini seçer.
9. Soğutucu Sistemi, sıcaklık algılayıcısından sıcaklık değerini alır ve ekrana sıcaklık değeri yazdırılır.

Alternatif Olay Akışı:

- A. Kullanıcı Adı doğrulanmadı.
 - 3 den az kez yanlış ise yeniden gir.
 - 3 kez yanlış girilmiş ise işlemi sonlandır ve çıkış yap.
- B. Yanlış şifre girildi.
 - 3 den az kez yanlış ise yeniden gir.
 - 3 kez yanlış şifre girilmiş ise işlemi sonlandır ve çıkış yap.

Özel Gereksinimler:

- Sıcaklık Algılayıcı çalışır durumda olmalıdır.

Kullanım Durumu: Soğutucunun Çalıştırılması

Hazırlayan: Adem Yılmaz

Sürüm: v1.0

Tarih: 19/04/2022

İlgili Aktörler: İnternet Kullanıcısı, Soğutucu Bilgi Sistemi

Giriş Koşulu:

- Kullanıcı sisteme giriş yapmış olmalıdır.
- Cihaz açık ve kullanılabilir durumda olmalıdır.

Çıkış Koşulu: Kullanıcı işlemi tamamlanmıştır.

Ana Olay Akışı(Ana Senaryo Başarılı)

1. Soğutucu sistemi, ekrana kullanıcıdan kullanıcı adını girmesini isteyen bir mesaj yazdırır.
2. Kullanıcı, tuş takımını kullanarak kullanıcı adını girer.
3. Kullanıcı adı doğrulaması için Soğutucu Bilgi Sistemine istek gönderilir.
4. Sistem isteği kabul eder ve ekrana kullanıcıdan şifre girilmesini isteyen mesaj gönderilir.
5. Kullanıcı, tuş takımını kullanarak şifreyi girer.
6. Kullanıcı doğrulaması için soğutucu sistemine istek gönderilir.
7. Soğutucu sistemi erişim isteğini kabul eder ve arayüze erişim sağlanır.
8. Kullanıcı, arayüzden tuş takımı yardımıyla soğukluk çalıştırma seçeneklerinden (açma veya kapama) birini seçer.
9. Soğutucu Sistemi, eyleyici çağırır ve gerekli olan işlemi(soğutucuyu açma veya kapama) yapmasını ister.
10. Soğutucu sistemi seçilen işleme göre soğutucuyu devreye sokar veya devre dışı bırakır.
11. Soğutucu sistemi, sıcaklık algılayıcısını çağırır.Sıcaklık algılayıcısı, soğutucu devrede ise sıcaklığı düşürür veya soğutucu devre dışı ise sıcaklığı oda sıcaklık düzeyine getirir.
12. Soğutucu açma işlemi seçilmiş ise, ekrana soğutucu devrede yazdırılır.
13. Soğutucu kapama işlemi seçilmiş ise, ekrana soğutucu devredışı yazdırılır.
14. Ortalama sıcaklık değeri ekrana yazdırılır.

Alternatif Olay Akışı:

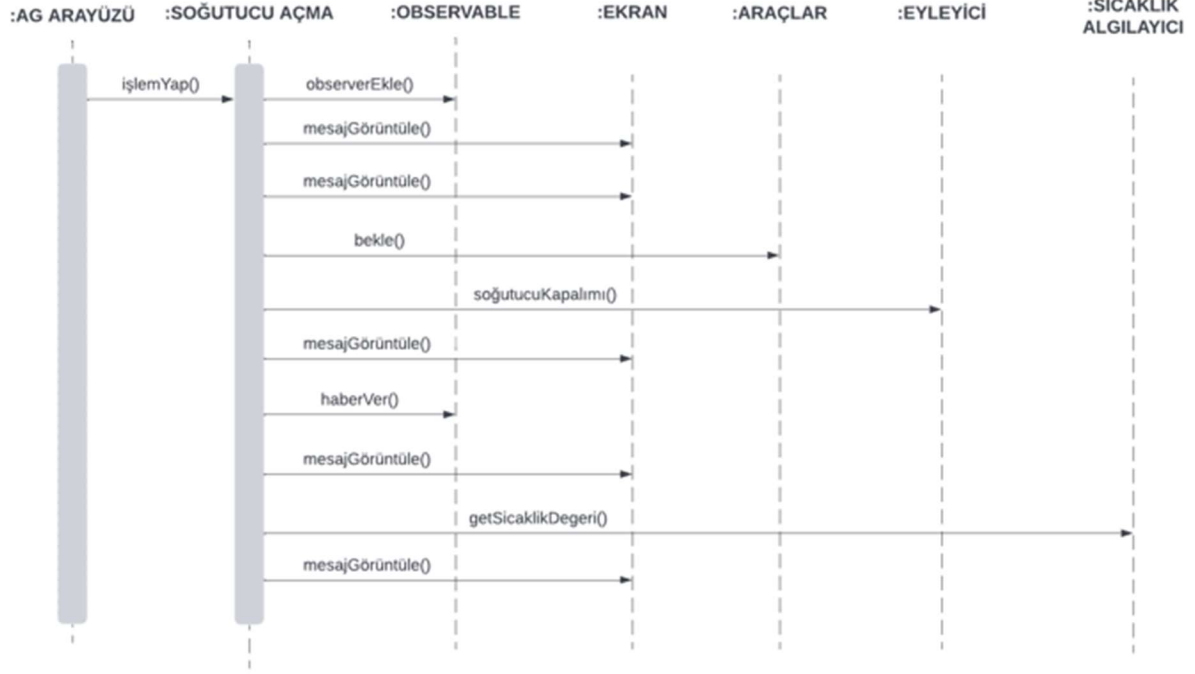
- A.** Kullanıcı Adı doğrulanmadı.
 - 3 den az kez yanlış ise yeniden gir.
 - 3 kez yanlış girilmiş ise işlemi sonlandır ve çıkış yap.
- B.** Yanlış şifre girildi.
 - 3 den az kez yanlış ise yeniden gir.
 - 3 kez yanlış şifre girilmiş ise işlemi sonlandır ve çıkış yap.
- C.** Soğutucu Açık(Soğutucu Açma İşlemi)
 - Ekrana soğutucunun açık durumda olduğu ve sıcaklık değeri yazdırılır.
 - İşlem sonlandırılır.
- D.** Soğutucu Kapalı(Soğutucu Kapama İşlemi)
 - Ekrana soğutucunun kapalı durumda olduğu ve sıcaklık değeri yazdırılır.
 - İşlem sonlandırılır.

Özel Gereksinimler:

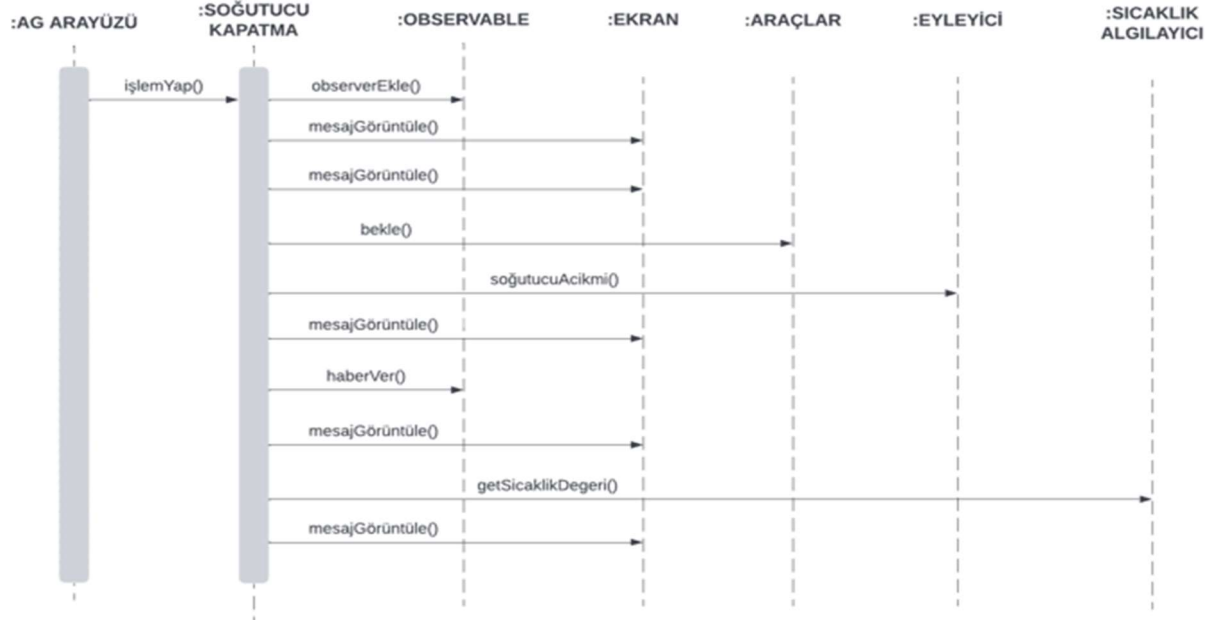
- Sıcaklık Algılayıcı çalışır durumda olmalıdır.

SIRALAMA ŞEMALARI

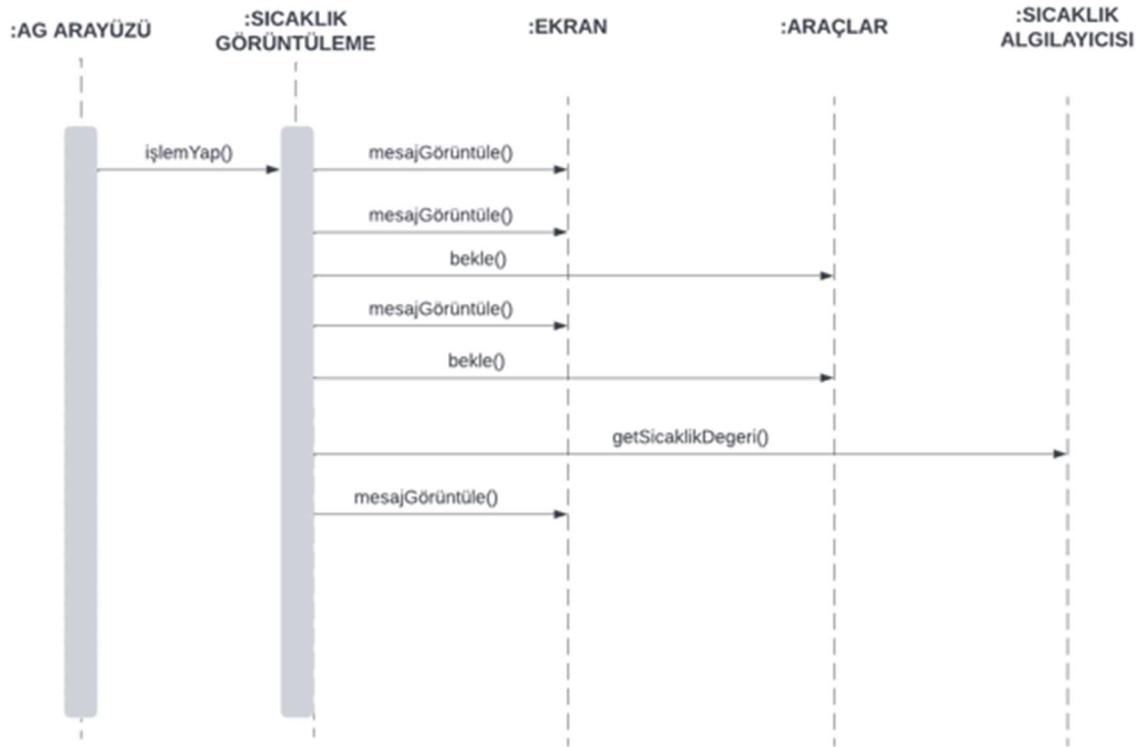
Soğutucunun Çalıştırılması



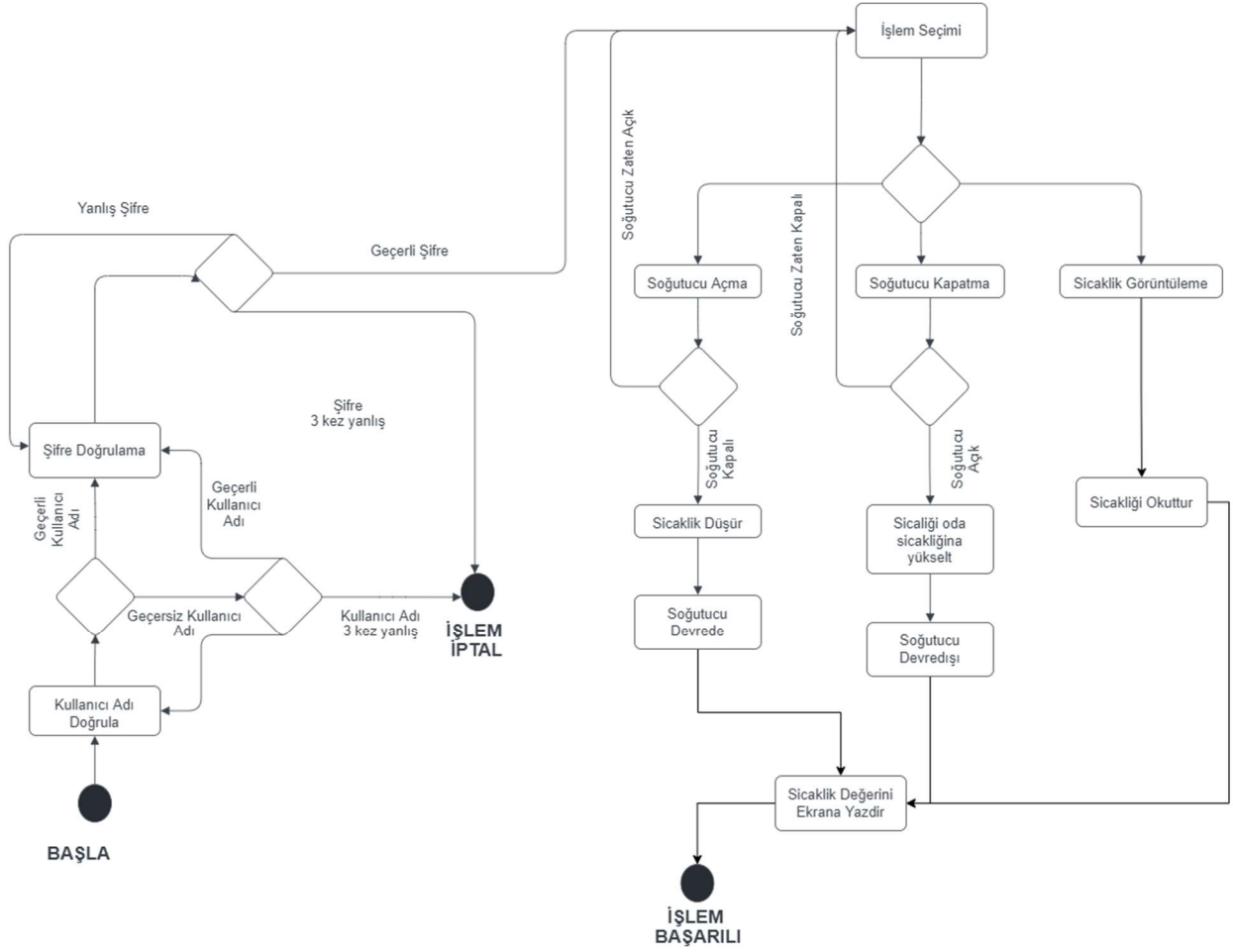
Soğutucunun Kapatılması



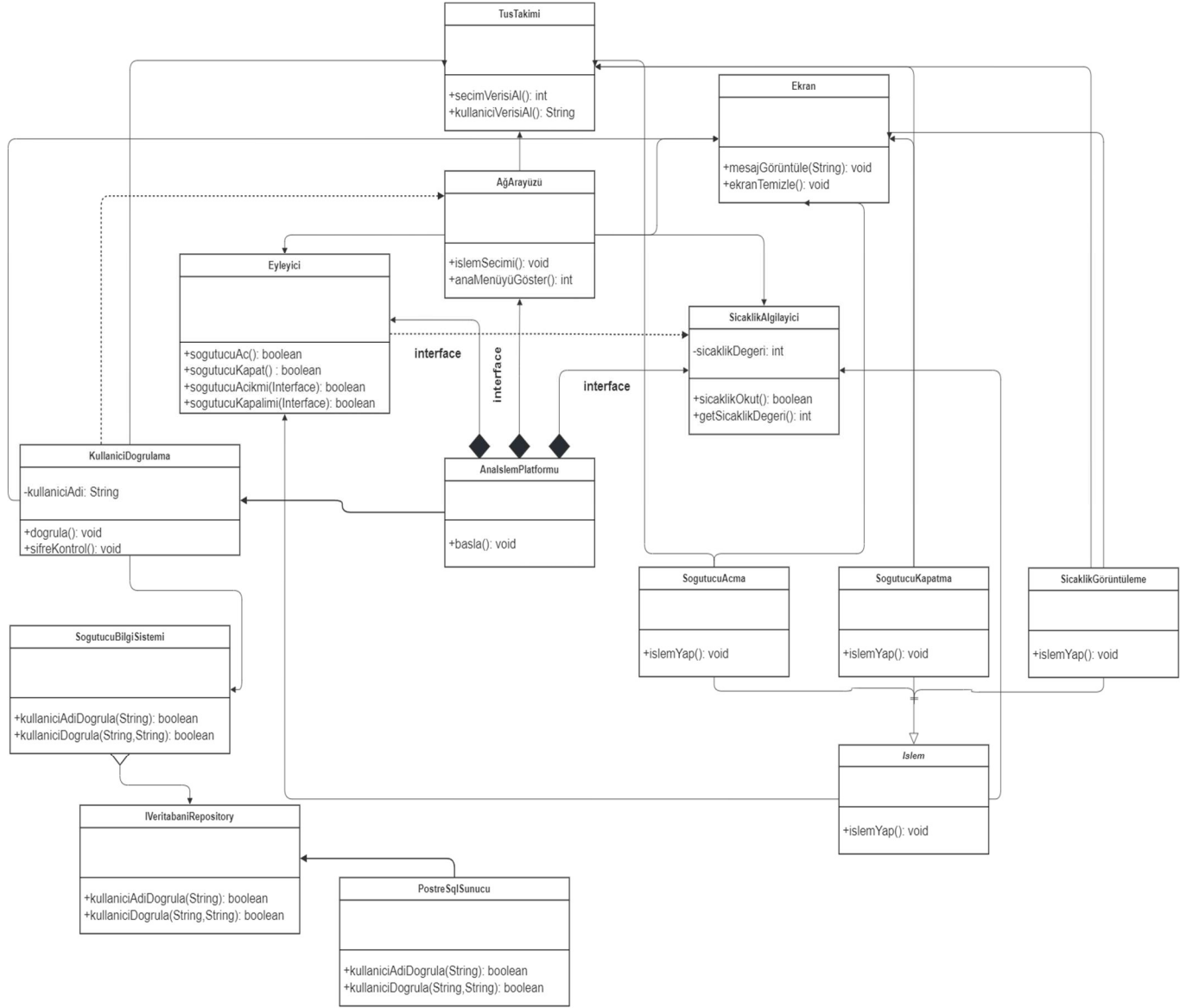
Sıcaklığın Görüntülenmesi



ETKİNLİK ŞEMASI



SINIF ŞEMASI

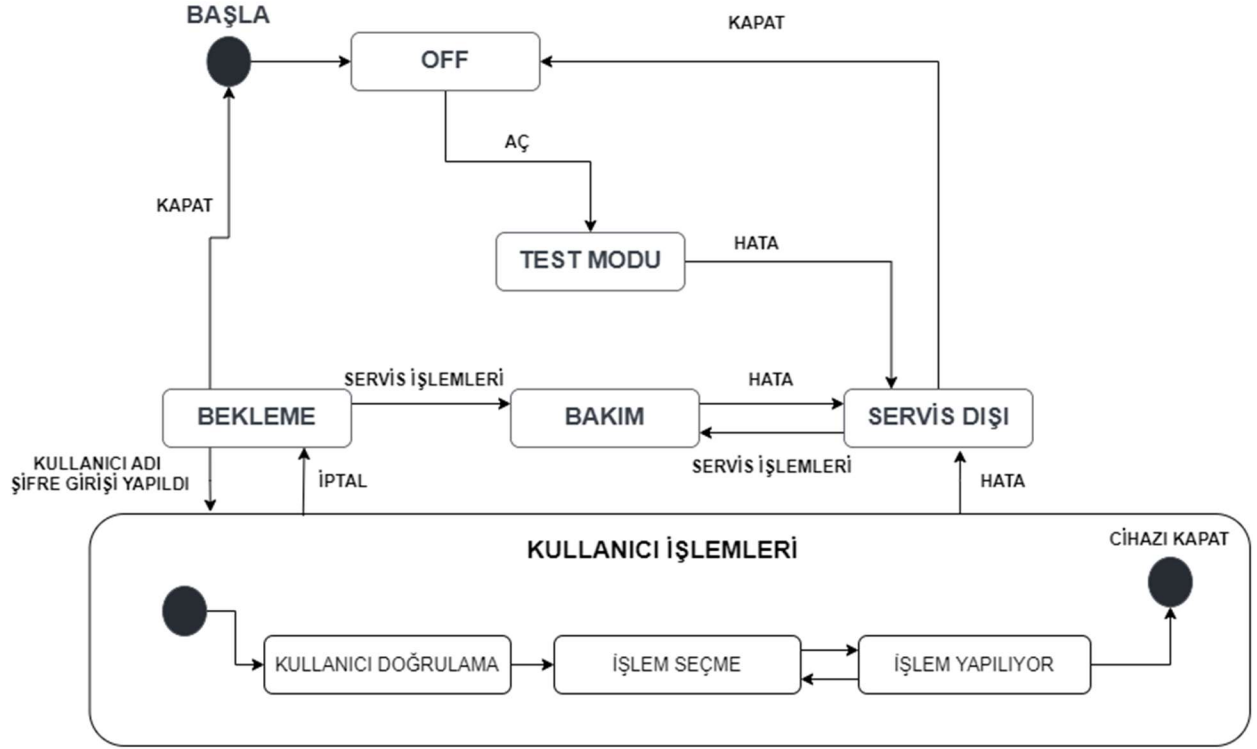


CRC KARTLARI

Soğutucu Açma(Soğutucu Açma işlemini gerçekleştirir)	
SORUMLULUK	İŞBİRLİĞİ YAPILAN SINIF
Mesaj yazdırma	Ekran
Soğutucu Kontrolü(Açık mı, Kapalı mı?)	Eyleyici
Sıcaklık düşürme	Sicaklik Algilayici, Observer
Soğutucu açma	kendisi
Sıcaklık alma	Sicaklik Algilayici

Kullanici Dogrulama(Kullanici doğrulama ve anamenüye yönlendirme işlemini gerçekleştirir)	
SORUMLULUK	İŞBİRLİĞİ YAPILAN SINIF
Mesaj yazdırma	Ekran
Kullanici adi ve şifre alma	Tuş takımı
Kullanici adi ve şifre doğrulama	SoğutucuBilgiSistemi(IVeritabanıRepository, PostresqlSunucu)
Anamenüye yönlendirme	kendisi

DURUM MAKİNESİ DİYAGRAMI



UYGULAMA EKRAN GÖRÜNTÜLERİ

Hatalı Kullanıcı

Hoşgeldiniz...
Kullanici Adinizi giriniz:
adem
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz kullanıcı adi hatali..
Tekrar Kullanici Adi Giriniz:
adksaljdklsajlda
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz kullanıcı adi hatali..
Tekrar Kullanici Adi Giriniz:
adslşakşkdlsşa
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
3 den fazla hatali kullanıcı adi girdiniz.Cikis yapiliyor...

Hatalı Şifre

Hoşgeldiniz...
Kullanici Adinizi giriniz:
adem28
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Sifre giriniz:
dsadasddd
Sifre Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz Sifre Hatali..
Tekrar Sifre Giriniz:
12345
Sifre Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz Sifre Hatali..
Tekrar Sifre Giriniz:
456445
Sifre Dogrulaniyor...
Veritabanına bağlandı!
3 den fazla hatali sifre girdiniz.Cikis yapiliyor...

Başarılı Doğrulama

Hoşgeldiniz...

Kullanıcı Adınızı giriniz:

adem

Kullanıcı Adı Doğrulaniyor...

Veritabanına bağlandı!

Girdiğiniz kullanıcı adı hatalı..

Tekrar Kullanıcı Adı Giriniz:

adem28

Kullanıcı Adı Doğrulaniyor...

Veritabanına bağlandı!

Sifre giriniz:

1245

Sifre Doğrulaniyor...

Veritabanına bağlandı!

Girdiğiniz Sifre Hatalı..

Tekrar Sifre Giriniz:

123

Sifre Doğrulaniyor...

Veritabanına bağlandı!

Giris Başarılı...adem28

Ana Menu

1-Sogutucu Ac

2-Sogutucu Kapat

3-Sicaklık Görüntüle

4-Cikis Yap

Seciminiz:

Soğutucunun Çalıştırılması

```
*****
Ana Menu
1-Sogutucu Ac
2-Sogutucu Kapat
3-Sicaklik Görüntüle
4-Cikis Yap
Seciminiz:
*****
1
Sogutucu acma islemini sectiniz.
Sogutucu acma islemi baslatiliyor...
Sicaklik düşürülüyor...
Sogutucu Devrede !!!
Ortalama Sicaklik Degeri: -7
Ana Menüye Dönmek için '5' i tuşlayınız.
|
```

Soğutucunun Kapatılması

```
Ana Menüye Dönmek için '5' i tuşlayınız.
5
*****
Ana Menu
1-Sogutucu Ac
2-Sogutucu Kapat
3-Sicaklik Görüntüle
4-Cikis Yap
Seciminiz:
*****
2
Sogutucu kapama islemini sectiniz.
Sogutucu kapama islemi baslatiliyor...
Sicaklik oda sicakligina yükseltiliyor...
Sogutucu Devredışı !!!
Ortalama Sicaklik Degeri: 13
Ana Menüye Dönmek için '5' i tuşlayınız.
|
```

Sıcaklığın Görüntülenmesi

2

Sogutucu kapama islemini sectiniz.
Sogutucu kapama islemi baslatiliyor...
Sicaklik oda sicakligina yükseltiliyor...
Sogutucu Devrededişi !!!
Ortalama Sicaklik Degeri: 13
Ana Menüye Dönmek için '5' i tuşlayınız.

5

Ana Menu

- 1-Sogutucu Ac
- 2-Sogutucu Kapat
- 3-Sicaklik Görüntüle
- 4-Cikis Yap

Seciminiz:





3

Sicaklik görüntüleme islemini sectiniz.
Sicaklik görüntüleme islemi baslatiliyor...
Algilayici'dan sicaklik degeri aliniyor...
Ortalama Sicaklik Degeri: 13
Ana Menüye Dönmek için '5' i tuşlayınız.

,

VERİTABANI EKRAN GÖRÜNTÜLERİ

Kullanıcılar Tablosu ve Verileri

Data Output	Explain	Messages	Notifications
	id [PK] integer 	kullaniciadi character varying (50) 	sifre character varying (50) 
1	1	adem28	123
2	2	kullanici	789
3	3	kullanici2	123456

SQL Kodları

Tablo Oluşturma ve ilgili tabloya veri girilmesi

```
CREATE TABLE "Users" (  
    "UserId" SERIAL,  
    "UserName" VARCHAR(50) NOT NULL,  
    "Password" VARCHAR(50) NOT NULL,  
    CONSTRAINT "userIdPK" PRIMARY KEY("UserId"),  
    CONSTRAINT "userNameUnique" UNIQUE("UserName")  
);
```

```
INSERT INTO "Users"("UserName","Password") VALUES ('adem28','123')  
INSERT INTO "Users"("UserName","Password") VALUES ('kullanici','789')  
INSERT INTO "Users"("UserName","Password") VALUES ('kullanici2','123456')
```

Kullanıcı Adı ve Kullanıcı Doğrulama Fonksiyonlarının Oluşturulması

```
15 CREATE FUNCTION kullaniciadidogrula(_userName character varying) returns integer
16 AS
17 $$
18 BEGIN
19     IF (select count(*) from "Users" where kullaniciadi=_userName)>0 THEN
20         return 1;
21     ELSE
22         return 0;
23     END IF;
24 END;
25 $$
26 LANGUAGE "plpgsql";
27
28 CREATE FUNCTION kullanicidogrula(_userName character varying,_sifre character varying) returns integer
29 AS
30 $$
31 BEGIN
32     IF (select count(*) from "Users" where kullaniciadi=_userName and sifre=_sifre)>0 THEN
33         return 1;
34     ELSE
35         return 0;
36     END IF;
37 END;
38 $$
39 LANGUAGE "plpgsql";
40
41 select * from kullaniciadidogrula('adem28');
42 select * from kullanicidogrula('adem28','123');
```

Data Output Explain Messages Notifications

kullanicidogrula
integer

1	1
---	---

DEPENDENCY INVERSION İLKESİ

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

Dependency Inversion ilkesinde ki temel amaç, bir sınıf içerisinde başka bir sınıfı kullanıyorsanız kısacası kullanılan sınıf direkt olarak somut haliyle değil, onu soyut haliyle kullanmaktır.

Burada Soğutucu Bilgi Sistemi, bir veritabanı kullanarak kullanıcı doğrulaması yapacaktır.

Sistem de birden fazla veritabanı olduğunu varsayalım. Dolayısıyla ben sistemdeki herhangi bir veritabanını kullanarak doğrulama yapabilirim. Her bir veritabanının ise kendine özgü ayrı kodları komutları vardır. Bundan dolayı benim bu veritabanılarını kolay olarak kullanabilmek adına bir interface tanımlayarak bunları bir interface altında toplayabilirim ve istediğim veritabanını bu interface yardımıyla kolayca kullanabilirim. Interface kullanmasaydım ben bu veritabanı sınıfının somutunu kullanmış olacaktım ve bu istenilen bir durum değildir. Dolayısıyla ben bu sınıfların somutunu değilde soyutunu kullanmak istersem veritabanı sınıfları ile Soğutucu Bilgi Sistemi sınıfı arasına bir interface koymam yeterlidir. Bu interface ise aşağıda görüldüğü üzere IVeritabanıRepository dir. Dolayısıyla artık ben IVeritabanıRepository interface ini kullanarak ilgili veritabanı sınıfın soyutunu kullanabilirim ve diğer veritabanı sınıflarına olan bağımlılığımı azaltabilirim.

```
public class SoğutucuBilgiSistemi implements ISoğutucuBilgiSistemi {
    private IVeritabanıRepository veritabanıRepository;
}
public SoğutucuBilgiSistemi(IVeritabanıRepository veritabanıRepository){
    this.veritabanıRepository=veritabanıRepository;
}

@Override
public boolean kullanıcıAdıDogrula(String kullanıcıAdi) {
    if(veritabanıRepository.kullanıcıAdıDogrula(kullanıcıAdi)) { return true; }
    return false;
}

@Override
public boolean kullanıcıDogrula(String kullanıcıAdi, String sifre) {
    if(veritabanıRepository.kullanıcıDogrula(kullanıcıAdi,sifre)){ return true; }
    return false;
}
```

```
public interface IVeritabaniRepository {  
    public boolean kullaniciAdiDogrula(String kullaniciAdi);  
    public boolean kullaniciDogrula(String kullaniciAdi,String sifre);  
}
```

```
public class PostreSqlSunucu implements IVeritabaniRepository {
```

```
    private Connection baglan() {  
        Connection conn=null;  
        try {  
            conn = DriverManager.getConnection( url: "jdbc:postgresql://localhost:5432/veritabani",  
                user: "postgres", password: "123");  
  
            System.out.println("Veritabanına bağlandı!");  
        } catch (Exception e) {  
            System.out.println("Bağlantı girişimi başarısız!");  
            e.printStackTrace();  
        }  
        return conn;  
    }  
  
    @Override  
    public boolean kullaniciAdiDogrula(String kullaniciAdi) {  
        Connection conn=this.baglan();  
        try {  
            CallableStatement stmt = conn.prepareCall( sql: "{?= call kullaniciadidogrula(?)}" );  
            stmt.setString( parameterIndex: 2,kullaniciAdi);  
            stmt.registerOutParameter( parameterIndex: 1, Types.INTEGER);  
            stmt.execute();  
            conn.close();  
            if(stmt.getInt( parameterIndex: 1)==1)  
                return true;  
        }  
    }  
}
```

FACTORY METHOD DESENİ

Factory Method deseni, yazılım ortamında birden çok aynı özelliği gösterebilecek sınıflar için gerekli nesne üretiminin kalıtım yolu ile kalıtım yapılan nesne tarafından yapılmasını sağlamaktır.

Geliştirmiş olduğum uygulamada factory method desenine benzer bir yapıyı, aşağıda ekran görüntülerinde görmüş olduğunuz üzere interface olarak değilde, bir abstract sınıf kullanarak oluşturdum.

Burada Abstract sınıfım olan Islem sınıfı fabrika sınıfımdır , factory method um ise islemyap() yöntemidir. Bu sınıftan miras alan sınıflar ise, SogutucuAcma, sogutucu kapatma ve sicaklik görüntüleme sınıflarıdır. Görmüş olduğunuz üzere her bir sınıfın kendine ait islemyap() yöntemi var. Dolayısıyla ben factory method kullanıp kalıtım alınan sınıftan nesne üretebiliyorum ve alttaki resimde görüldüğü üzere bu nesne vasıtasıyla işlem yapabiliyorum.

```
secim=anaMenüyüGöster();
switch (secim){
    case sogutucuAcma:
        if(eyleyici.SogutucuAc()){
            Islem sogutucuAcma = new SogutucuAcma(ekran,eyleyici,sicaklikAlgilayici);
            sogutucuAcma.islemYap();
            break;
        }
    case sogutucuKapatma:
        if(eyleyici.SogutucuKapat())
        {
            Islem sogutucuKapatma= new SogutucuKapatma(ekran,eyleyici,sicaklikAlgilayici);
            sogutucuKapatma.islemYap();
            break;
        }
    case sicaklikGörüntüleme:
        if(sicaklikAlgilayici.SicaklikOkut())
        {
            Islem sicaklikGörüntüleme=new SicaklikGörüntüleme(ekran,sicaklikAlgilayici);
            sicaklikGörüntüleme.islemYap();
            break;
        }
    ...
}
```

```

public abstract class Islem extends Observable {
    protected final IEkran ekran;
    protected final IEyleyici eyleyici;
    protected final ISicaklikAlgilayici sicaklikAlgilayici;

    protected Islem(IEkran ekran, IEyleyici eyleyici, ISicaklikAlgilayici sicaklikAlgilayici) {
        this.ekran = ekran;
        this.eyleyici = eyleyici;
        this.sicaklikAlgilayici = sicaklikAlgilayici;
    }

    public void observerEkle(){ ekle((Observer) sicaklikAlgilayici); };
    public abstract void islemYap();
}

```

```

public class SogutucuAcma extends Islem {

    public SogutucuAcma(IEkran ekran, IEyleyici eyleyici, ISicaklikAlgilayici sicaklikAlgilayici) {
        super(ekran, eyleyici, sicaklikAlgilayici);
    }

    @Override
    public void islemYap() {
        observerEkle();
        ekran.mesajGoruntule("Sogutucu acma islemini sectiniz.");
        ekran.mesajGoruntule("Sogutucu acma islemi baslatiliyor...");
        Araclar.bekle( sure: 2000);
        if(eyleyici.SogutucuKapalimi(sicaklikAlgilayici))
        {
            ekran.mesajGoruntule("Sicaklik düşürülüyor...");
            haberVer( deger: "sicaklikdüşür");
            Araclar.bekle( sure: 2000);
            ekran.mesajGoruntule("Sogutucu Devrede !!!");
            Araclar.bekle();
        }
    }
}

```

OBSERVER TASARIM DESENİ

Observer tasarım deseni, birden fazla nesneyi takip ettikleri başka bir nesnede gerçekleşen olaylarla ilgili bilgilendirmeyi sağlayan bir abonelik mekanizması oluşturmayı amaçlar.

Gerçekleştirdiğim uygulamada observer tasarım desenini aşağıdaki ekran görüntülerinde görüldüğü üzere Soğutucu açma, soğutucu kapatma gibi durumlarda sıcaklık seviyesinin tekrardan güncellenmesi (sıcaklığın düşürülmesi veya oda sıcaklığına yükseltilmesi) için kullandım.

Burada observer'ım sıcaklık algılayıcısı , observable larım ise soğutucu açma ve soğutucu kapama durumlarıdır.

```
public interface Observer {  
    void update(String deger);  
}
```

Birden fazla sıcaklık algılayıcısı olabilir ve bunların her birine güncelleme yapmak gereklidir. Dolayısıyla bu durumu dikkate alarak bir observer listesi tanımladım.

```
public abstract class Observable {  
    private List<Observer> observerList;  
  
    public Observable() { observerList = new ArrayList<>(); }  
  
    public void ekle(Observer observer) { observerList.add(observer); }  
  
    public void cikar(Observer observer) { observerList.remove(observer); }  
  
    public void haberVer(String deger){  
        for (Observer observer : observerList) {  
            observer.update(deger);  
        }  
    }  
}
```


Observer in update durumunda yapacağı işlemler aşağıda görülmek üzere sıcaklık düşürme ve oda sıcaklığı yöntemlerinde belirtilmiştir.

```
public class SıcaklıkAlgılayıcı implements ISıcaklıkAlgılayıcı, Observer {
    Random rnd = new Random();
    private int sıcaklıkDegeri;

    public SıcaklıkAlgılayıcı() { setSıcaklıkDegeri(rnd.nextInt( origin: 10, bound: 25)); }
    public int getSıcaklıkDegeri() { return sıcaklıkDegeri; }

    private void setSıcaklıkDegeri(int sıcaklıkDegeri) { this.sıcaklıkDegeri = sıcaklıkDegeri; }

    private void odaSıcaklığınaAyarla() { setSıcaklıkDegeri(rnd.nextInt( origin: 10, bound: 35)); }

    private void sıcaklıkDusur() { setSıcaklıkDegeri(rnd.nextInt( origin: -10, bound: 0)); }

    @Override
    public boolean SıcaklıkOkut() { return true; }

    @Override
    public void update(String ifade) {
        if (ifade.equalsIgnoreCase( anotherString: "sıcaklıkdüşür"))
            sıcaklıkDusur();
        else if (ifade.equalsIgnoreCase( anotherString: "odasıcaklığı"))
            odaSıcaklığınaAyarla();
    }
}
```

Soğutucu Açılırken sıcaklığın düşürülmesi durumudur.

```
public class SoğutucuAcma extends Islem {

    public SoğutucuAcma(IEkran ekran, IEyleyici eyleyici, ISıcaklıkAlgılayıcı sıcaklıkAlgılayıcı) {
        super(ekran, eyleyici, sıcaklıkAlgılayıcı);
    }

    @Override
    public void islemYap() {
        observerEkle();
        ekran.mesajGoruntule("Soğutucu acma islemini sectiniz.");
        ekran.mesajGoruntule("Soğutucu acma islemi baslatılıyor...");
        Araclar.bekle( sure: 2000);
        if(eyleyici.SogutucuKapalimi(sıcaklıkAlgılayıcı))
        {
            ekran.mesajGoruntule("Sıcaklık düşürülüyor...");
            haberVer( deger: "sıcaklıkdüşür");
            Araclar.bekle( sure: 2000);
            ekran.mesajGoruntule("Soğutucu Devrede !!!");
            Araclar.bekle();
        }
    }
}
```