NESNE YÖNELİMLİ ANALİZ VE TASARIM PROJE ÖDEVİ

Soğutma Sistemi



Ekrem Eren Yılmaz G181210039 2. Öğretim A grubu ekrem.yilmaz4@ogr.sakarya.edu.tr

HOSGELDINIZ

Kullanici Adinizi Giriniz: admin

admin

Sifrenizi Giriniz: admin

1. Giriş Ekranı

Veritabanına bağlanarak kullanıcı adı ve şifre kontrolü yapılan doğrulama ekranı.

2. Arayüz

-5, 20 derece arası belirlenen sıcaklık değerinin görüntülenmesi

3. Soğutucu Açma ve Kapatma

Soğutucunun soğutucunun açılması durumunda soğutucunun açık veya kapalı olduğu durumda verdiği çıktılar,

```
*****************

*** MENU

*** 1 - Sicaklik Goruntule

*** 2 - Sogutucuyu ac

*** 3 - Sogutucuyu kapat

*** 4 - Cikis yap

*** Seciminiz: 2

Sogutucu zaten acik
```

```
***************

*** MENU

*** 1 - Sicaklik Goruntule

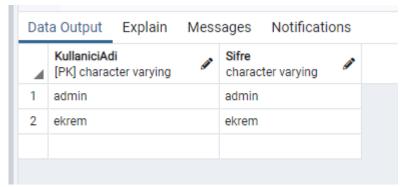
*** 2 - Sogutucuyu ac

*** 3 - Sogutucuyu kapat

*** 4 - Cikis yap

*** Seciminiz: 3

Sogutucu zaten kapali
```



4. Veritabanı

PostreSql veritabanı içerisinde yer alan kullanıcılara ait kullanıcı adı ve şifreleri

Dependency Inversion

Dependency inversion tasarım ilkesi eyleyici nesnesinde kullanıldı.

Kullanıcı 1 durumunu seçtiğinde Motor1 nesnesi, 2 seçildiğinde ise motor2 nesnesi çağırıldı.

```
ekran.yazdir("Sogutucu kacinci kademede acilsin? 1-2: ");
int sec = tus.sayiOku();
if(sec == 1)
    eyleyici = new Motor1();
else
    eyleyici = new Motor2();
```

Ardından eyleyici nesnene ait soğutucu aç içerisine sıcaklık nesnesi gönderilerek motor1 nesnesi açıldığında sıcaklık içerisinde bulunan soğutucu sıcaklığı kademe 1 de '-5' kademe 2 de ise '-10' derece düşürüldü.

```
eyleyici.sogutucuAc(sicaklik);
sogutucu_durum = true;
```

Builder

Builder nesnesi, kullanıcı nesnesi oluştururken Kullanıcı içerisinde kullanıldı. Herhangi bir Kullanıcı kurucusu kullanmadan builder içerisindeki set methodu ile kullanıcıya ait kullanıcı adı ve şifre atamasını gerçekleştirildi.

```
while(rs.next())

if (rs.getString("KullaniciAdi").equals(kullaniciAdi) && rs.getString("Sifre").equals(sifre))

{
    kullanici = new Kullanici.Builder().set(kullaniciAdi, sifre).build();
}
```

Observer

Observer tasarım ilkesi, eyleyici içerisinde soğutucunun açılması ve kapatılması durumunda sıcaklık nesnesini soğutucu aç ve soğutucuyu kapat nesneleri içerisine gönderilmesi esnasında kullanıldı.

```
public void sogutucuAc(ISicaklikAlgilayici sicaklik);
public void sogutucuKapat(ISicaklikAlgilayici sicaklik);

public void sogutucuKapat(ISicaklikAlgilayici sicaklik)
{
    sicaklik.update(10);
    System.out.println("Sogutucu kapatildi");
}

eyleyici.sogutucuAc(sicaklik);

public void update(int der)
    { sicaklik += der; }
```

Github linki:

- https://github.com/ekremrn/NesneYonelimliAnalizveTasarim-Proje-Odevi

Video linki:

https://www.youtube.com/watch?v=EkcrKgFOBR8