



SAKARYA ÜNİVERSİTESİ  
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
NESNE YÖNELİMLİ ANALİZ VE TASARIM DERSİ

PROJE RAPORU

ALEYNA ELİF ÖZKAN

B201210387

2A

[aleyna.ozkan1@ogr.sakarya.edu.tr](mailto:aleyna.ozkan1@ogr.sakarya.edu.tr)

## Kullanıcı doğrulaması

Kullanıcıdan kullanıcı adı ve parola girilmesi istenir. Sistem veritabanına bağlanır ve kullanıcının bilgilerini doğrular. Eğer girilen bilgiler doğruysa giriş yapılır ve menü gözükür.

```
KULLANICI GİRİŞ ARAYÜZÜ
Kullanıcı Adı ve Parola giriniz:
Kullanıcı Adı: aleyna
Şifre: dora
Veritabanına bağlandı.
Giriş başarılı.

MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:|
```

Eğer veritabanında böyle bir kullanıcı bulunamadıysa hata mesajı bastırılıp programdan çıkılır.

```
KULLANICI GİRİŞ ARAYÜZÜ
Kullanıcı Adı ve Parola giriniz:
Kullanıcı Adı: aleyna
Şifre: dora1
Veritabanına bağlandı.
Böyle bir kullanıcı bulunamadı.
Process finished with exit code 0
```

## Sıcaklığın Görüntülenmesi

Eğer kullanıcı menüden 1 numaralı işlemi seçerse, görüntülemek istediği sıcaklık türünü seçmesi istenir. Bu işlem programın çalışma süresi boyunca sadece tek bir kez yapılır.

```
MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:1

Sıcaklık türünü seçiniz
-----
1. Celsius
2. Kelvin
Seçim: 1
Sıcaklık: -7,6 °C
```

```
MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:1

Sıcaklık türünü seçiniz
-----
1. Celsius
2. Kelvin
Seçim: 2
Sıcaklık: 50,28 °F
```

## Soğutucunun Açılması

Kullanıcı soğutucuyu açmak için 2. Seçeneği seçtiğinde eğer soğutucu kapalı durumdaysa “Soğutucu başarıyla açıldı.”, eğer soğutucu açık durumdaysa “Soğutucu Açık.” mesajı ekrana bastırılır.

```
MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:2
Soğutucu başarıyla açıldı.
```

```
MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:2
Soğutucu açık.
```

## Soğutucunun Kapatılması

Kullanıcı soğutucuyu kapatmak için 3. Seçeneği seçtiğinde eğer soğutucu açık durumdaysa “Soğutucu başarıyla kapatıldı.”, eğer soğutucu kapalı durumdaysa “Soğutucu Kapalı.” mesajı ekrana bastırılır.

```

MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:3
Soğutucu başarıyla kapatıldı.

```

```

MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:3
Soğutucu kapalı.

```

## Sistemden çıkılması

Kullanıcı 4. Seçeneği seçtiğinde sistemden çıkış yapılır.

```

MENU
-----
1.Sıcaklığı Göster
2.Soğutucuyu Aç
3.Soğutucuyu Kapat
4.Çıkış
Seçim:4

Çıkış yapılıyor...

Process finished with exit code 0

```

## Veritabanının Görüntüsü

Oluşturulan kullanıcının kullanıcı adı ve şifre bilgileri:

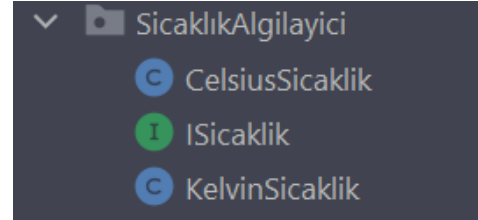
	Data Output	Explain	Messages	Notifications
	isim character varying		sifre character varying	
1	aleyna		dora	

## Dependency Inversion

S.O.L.I.D prensiplerinden biri olan Dependency Inversion Türkçeye bağımlılığın ters çevirilmesi prensibi olarak çevirilebilir. Bu prensibin amacı bir sınıfın, metodun veya özelliğin diğer sınıflara karşı olan bağımlılığını en aza indirmektir. Projenin sonradan değiştirilmesini kolaylaştırır. Ben bu prensibi Main.java adlı dosyada kullandım.

```
if (sicaklikTuru == 1)
    sicaklik = new CelsiusSicaklik();

else
    sicaklik = new KelvinSicaklik();
```



## Builder Tasarım Deseni

Oluşturulan sınıfın içinde çok fazla parametre alan obje varsa, farklı parametre kombinasyonlarına sahip birden çok constructor metodu varsa veya nesnenin oluşturulduktan sonra değişmez olması isteniyorsa builder tasarım deseni kullanılır. Ben bu tasarımı Kullanici.java dosyamın içinde kullandım.

```
public static class Builder
{
    private String isim;
    private String sifre;

    public Builder(){}

    public Builder set(String isim, String sifre)
    {
        this.isim = isim;
        this.sifre = sifre;
        return this;
    }

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

Kullanici(Builder builder)
{
    this.isim = builder.isim;
    this.sifre = builder.sifre;
}
```

## Observer

Observer çok sayıda nesneye, gözlemlenen nesnede gerçekleşen bir olayı bildirmek için kullanılan desendir. Projemde bu kısmı kullanamadım.

## Youtube Linki

[https://www.youtube.com/watch?v=DTuXzxned\\_8&ab\\_channel=aelynaelif%C3%B6zkan](https://www.youtube.com/watch?v=DTuXzxned_8&ab_channel=aelynaelif%C3%B6zkan)

## GitHub Linki

<https://github.com/aelynaelif/NesneYonelimliAnalizVeTasarimProje>