T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNE YÖNELİMLİ ANALİZ VE TASARIM ÖDEV RAPORU

Hasan Serhat POYRAZ

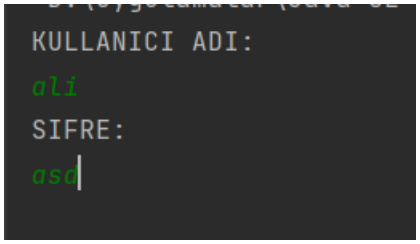
G191210055

2.Öğretim A Grubu

serhat.poyraz@ogr.sakarya.edu.tr

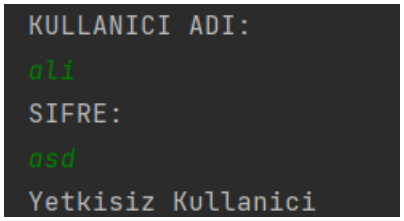
a. Kullanıcı doğrulama ekranı ve açıklaması

Program çalıştığında ilk olarak kullanıcı doğrulama ekranı açılıyor ve kullanıcıdan bilgilerini girmesi bekleniyor (1).

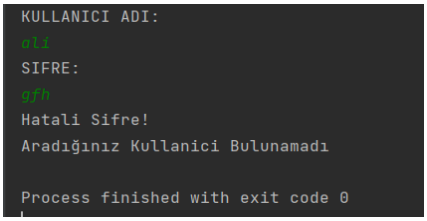


(1)

Kullanıcı bilgilerini şekildeki gibi girdikten sonra veritabanı üzerinden kullanıcının girmiş olduğu bilgiler doğrulanıyor.

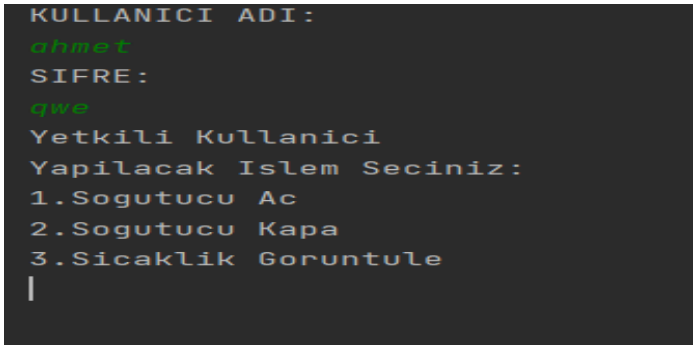


(2)



(3)

Şekilde (2) görüldüğü gibi giriş yapan kullanıcı eğer yetkisiz ise program uyarı veriyor ve sonlanıyor. Bunun yanı sıra eğer girilen kullanıcı adı ya da şifre hatalı ise (3) program ilgili hatayı ekrana yazıyor ve sonlanıyor.

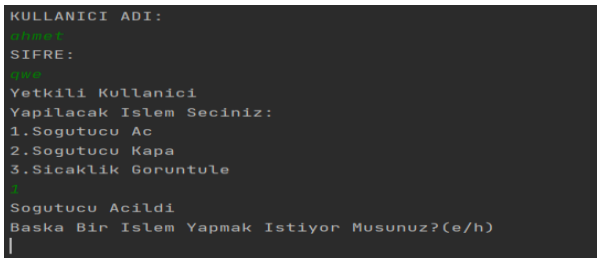


(4)

Şekildeki (4) gibi, veri tabanı üzerinden yapılan doğrulamanın ardından giriş yapan kullanıcı yetkili ise, kullanıcının yapabileceği işlemleri içeren menü ekrana çıkartılıyor.

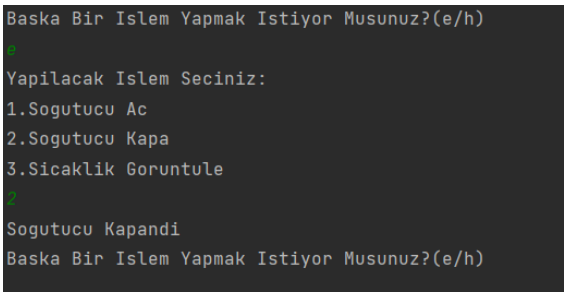
b. Sıcaklığın görüntülenmesi ve soğutucunun açılıp kapatılması işlemleri

Eğer kullanıcı soğutucu açma işlemini yani bir numaralı işlemi seçerse şekildeki (5) gibi ekrana ilgili mesaj çıkartılıyor ve kullanıcının başka bir işlem yapmak isteyip istemediği soruluyor.



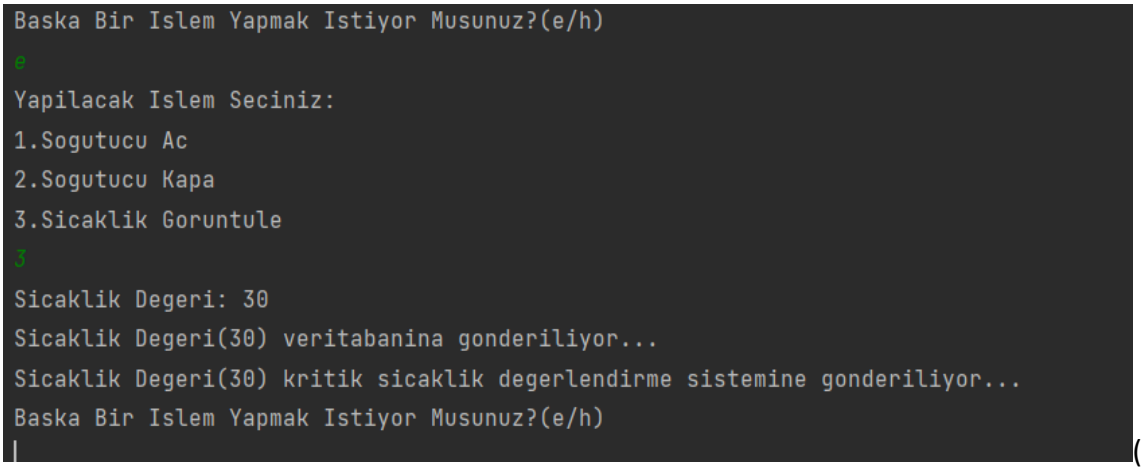
(5)

Eğer kullanıcı başka bir işlem daha yapmak ister ve soğutucuyu kapatmayı yani 2 numaralı işlemi seçerse şekildeki (6) gibi ekrana ilgili mesaj çıkartılıyor ve kullanıcının başka bir işlem yapmak isteyip istemediği soruluyor.



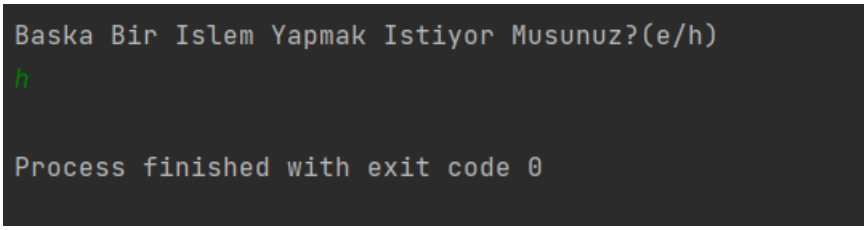
(6)

Eğer kullanıcı başka bir işlem daha yapmak ister ve sıcaklık görüntülemeyi yani 3 numaralı işlemi seçerse şekildeki (7) gibi ekrana sıcaklık değeri çıkartılıyor, ilgili observer işlemi (raporun ilerleyen kısmında detaylı anlatılacak) gerçekleştiriliyor ve kullanıcının başka bir işlem yapmak isteyip istemediği soruluyor.



(7)

Kullanıcı daha fazla işlem yapmak istemez ise şekildeki (8) gibi ‘h’ seçeneğini seçiyor ve program sonlandırılıyor.



(8)

c. Kullanıcı veritabanı

	id	yetki	kullaniciAdi	kullaniciSoyAdi	cinsiyet	sifre
1	1	✓	ahmet	demir	E	qwe
2	2	✓	ali	kara	E	asd
3	3	✓	arzu	acar	K	zxc

d. Dependency Inversion

Dependency inversion ilkesi iki sınıfın birbirine olan bağlılığını azaltmak yani zayıflatmak amacı ile kullanılan bir ilkedir. Üst seviye class ile onu kullanan düşük seviye class arasında oluşturulacak bir soyutlama katmanı aracılığı ile gerçekleştirilir. Ödevde soyutlama katmanı olarak interfaceleri kullandım. Örnek olması açısından ödevde kullandığım eleyici classında bu yapıyı kullandım, yani başka bir classda direkt olarak eleyici classından nesne üretip onun üzerinden işlem yapmak yerine eleyici classının interfacei üzerinden işlem yaparak eleyici classı ile bu classı kullandığım class arasındaki bağlantıyı zayıflatmış oldum. Bu yapıyı kullanarak ayrıca, henüz eleyici classının işlemlerini yapacağı metotların içeriğini yazmadan eleyici interface'i sayesinde eleyici classını başka classlarda kullanabildim bu da tasarım açısından büyük bir kolaylık sağladı.

e. Builder ve Observer desenleri

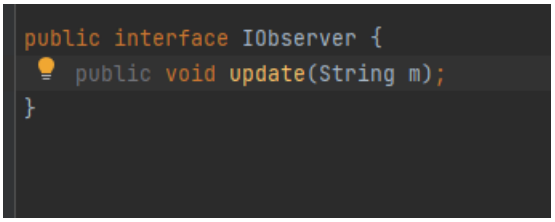
Builder: Nesne yönelimli programlamada sınıflar kullanılır. Sınıflardan nesneler üretiriz ve bunu yaparken kurucu metotlara ihtiyaç duyarız. Sınıfımızdaki değişken sayımız fazla olur ise birden fazla kurucu metoda ihtiyacımız olabilir. Kompleks sınıflarda her bir ihtimal için bir kurucu metot yazmak çok karışık ve zahmetli olacaktır bu noktada builder deseni bize büyük kolaylıklar sağlıyor. Ödevde ise kullanıcı bilgilerini içeren class içerisinde builder desenini kullandım. Builder desenini gerçeklerken ana class içerisinde static bir inner class kullandım (şekil 10) bu sayede class'dan nesne oluşturulurken hangi değerlerin isteğe bağlı olarak atanabileceğini belirledim. Ana classımın kurucusunu ise şekildeki (9) gibi tasarladım.

```
public KullaniciBilgileri(kullaniciBuilder builder)
{
    this.ad=builder.ad;
    this.soyAd=builder.soyAd;
    this.sifre=builder.sifre;
    this.yetki=builder.yetki;
}
```

(9)

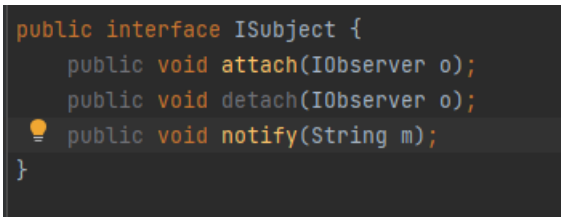
(10)

Observer: Bir nesne üzerindeki değişim bir veya birden çok nesneyi etkiliyor ise observer tasarım desenini kullanmak mantıklı olacaktır. Ödevde bu yapıyı, ölçülen sıcaklığı, kritik sıcaklığı kontrol eden bir mekanizmaya ve sıcaklıkların tutulduğu veritabanına mesaj göndermek için kullandım.



(11)

Şekildeki (11) interface ile observerımı tanımladım, mesajlar bu interfac üzerinden aktarılıyor. ISubject interface'i (şekil12) ise üyeler ile ilgili işlemleri; üye ekleme, üye çıkarma gibi işlemler ile ilgilenen interface'im.



(12)

Publisher classı (şekil 13) ile de ISubject interfaceimi genişlettim ve gerekli işlemleri yaptım.

```java
import java.util.ArrayList;
import java.util.List;

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) {
            try {
                Thread.sleep( millis: 1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            subscriber.update(mesaj);
        }
    }
}
```

(13)

Son olarak da iki adet classda (14,15) IObserver interface'imi imlement ettim yani bu iki classı birer gözlemci haline getirdim ve bu classlar artık notify() metodu ile gelen mesajı ilgili yerlere gönderiyorlar.

```java
public class KritikSicaklikDegerlendirme implements IObserver{
    @Override
    public void update(String sicaklik) {
        System.out.println("Sicaklik Degeri(" + sicaklik+") kritik sicaklik degerlendirme sistemine gonderiliyor...");
    }
}
```

(14)

```java
public class SicaklikDatabase implements IObserver {
    @Override
    public void update(String sicaklik) {
        System.out.println("Sicaklik Degeri(" + sicaklik+") veritabanina gonderiliyor...");
    }
}
```

(15)

f. Projenin kaynak kodları

https://github.com/serhatPoy/NYAT-PROJE

g. Projenin anlatıldığı video adresi

Youtube:
https://www.youtube.com/watch?v=WMtm7L62Qus