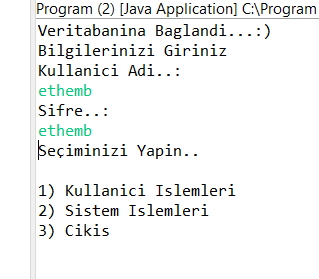
ETHEM BELKA ŞAHİN

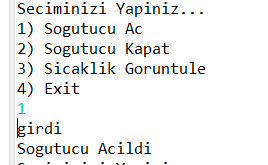
B181210061

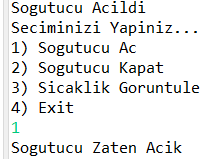
1. ÖĞRETİM A GRUBU

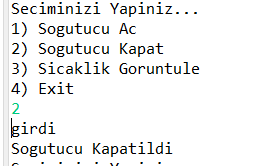
belka.sahin@ogr.sakarya.edu.tr

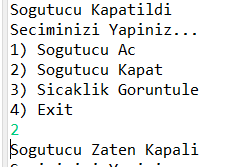
**1. KULLANICI DOĞRULAMA EKRANI VE AÇIKLAMASI**

**** Eğer veritabanına bağlanıra veritabanına bağlandı bilgisi ekrana yazdırılır. Daha sonra kullanıcı adını ve şifremizi girmemizi ister. Giriş başarılı ise menü ekrana gelir. Giriş başarılı değil ise tekrar giriş bilgilerini ister. **2. SICAKLIĞIN GÖRÜNTÜLENMESİ VE SOĞUTUCUNUN AÇILIP KAPANMASI**

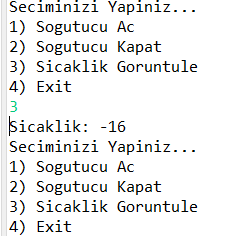
****Kullanıcı soğutucuyu aç seçeneğini seçer. Soğutucu açıldı bilgisi ekrana yazdırılır.

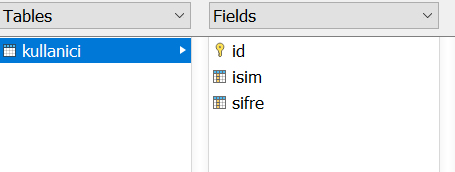
**** Kullanıcı soğutucu aç seçeneğini seçer. Soğutucu açık olduğu için soğutucu zaten açık bilgisi ekrana yazdırılır.

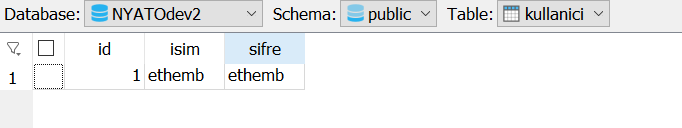
****Kullanıcı soğutucuyu kapat seçeneğini seçer. Soğutucu kapatıldı bilgisi ekrana yazdırılır.

****

Kullanıcı soğutucuyu kapat seçeneğini seçer. Soğutucu kapalı olduğu için soğutucu zaten kapalı bilgisi ekrana yazdırılır.

****Kullanıcı sıcaklığı görüntüle seçeneğini seçer. Random olarak sıcaklık ekrana yazdırılır. -16 yazdırıldı. **3. VERİTABANI GÖRÜNTÜSÜ**

****

****

**4. DEPENDDENCY INVERSION**

Dependency Inversion kodun belirli bir algoritmaya göre yazılmasını ve daha sonra eklemeler yaptığımızda daha önceki kodlar üzerinde bir değişim yapmamamızı sağlar. Proje içerisinde veritabanı, kullanıcı ve sistem paketleri içerisinde kullanılmıştır. Veritabanındaki kullanımı veritabanı adında bir interface oluşturulması ve kullanılacak veritabanının bu interfaceden implement edilmesi ile sağlanmıştır. Postgresql adındaki sınıfta bu interfacein method gövdeleri oluşturulmuştur. Daha sonra mysql, acces gibi farklı veritabanları eklendiğinde sisteme sadece o veritabanı ile ilgili class oluşturulup veritabanı interfaceini implement etmesi daha sonra da method gödelerinin yazılması yeterli olacaktır. Sistem paketi içerisinde oluşturduğum IIslem interfacei ise bu paket için kullanım kolaylığı sağlamaktadır. Soğutucu açma, kapama ve sıcaklık görüntüle işlemleri bu interfaceden implement edilmiştir. Daha sonra ekleyeceğim herhangi bir işlem için sadece bir sınıf oluşturup IIslemden implement edilmesi yeterli olacaktır.

**5. BUİLDER VE OBSERVER**

Builder’ı kullanıcı içerisinde kullanıcı bilgilerini alırken kullandım. Kullanıcı sınıfının içerisinde bir builder sınıfı oluşturdum ve kullanıcı adı ve şifresini builder sınıfı içerisinde ayrı methodlarda aldım. Daha sonra kullanıcı sınıfından nesne oluşturduğum zaman kullanıcı isim ve şifresini builder aracılığı ile adım.

Observer için bir paket oluşturdum ve Iobserver adında bir interface, Observerable adında da bir sınıf oluşturdum. Oluşturduğum obseverable sınıfının içinde observerları tuttuğum bir arraylist oluşturdum. Ve her observerablenin birden fazla olan gözlemvileri için bildirim yolla adında oluşturduğum method ile gözlenen bilgileri o gözleneni gözleyen tüm gözlemcilere ilettim.

**video linki :** [**https://www.youtube.com/watch?v=lSIZrmzX4ms**](https://www.youtube.com/watch?v=lSIZrmzX4ms)

**ETHEM BELKA ŞAHİN**

**B181210061**